Innovation, Sciences et
Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and
Economic Development Canada
Canadian Intellectual Property Office

CA 2837414 C 2022/12/13

(11)(21) **2 837 414**

(12) BREVET CANADIEN
CANADIAN PATENT

(13) **C**

(22) **Date de dépôt/Filing Date:** 2013/12/16

(41) **Mise à la disp. pub./Open to Public Insp.:** 2014/06/14

(45) **Date de délivrance/Issue Date:** 2022/12/13

(30) **Priorité/Priority:** 2012/12/14 (US US 61/737,726)

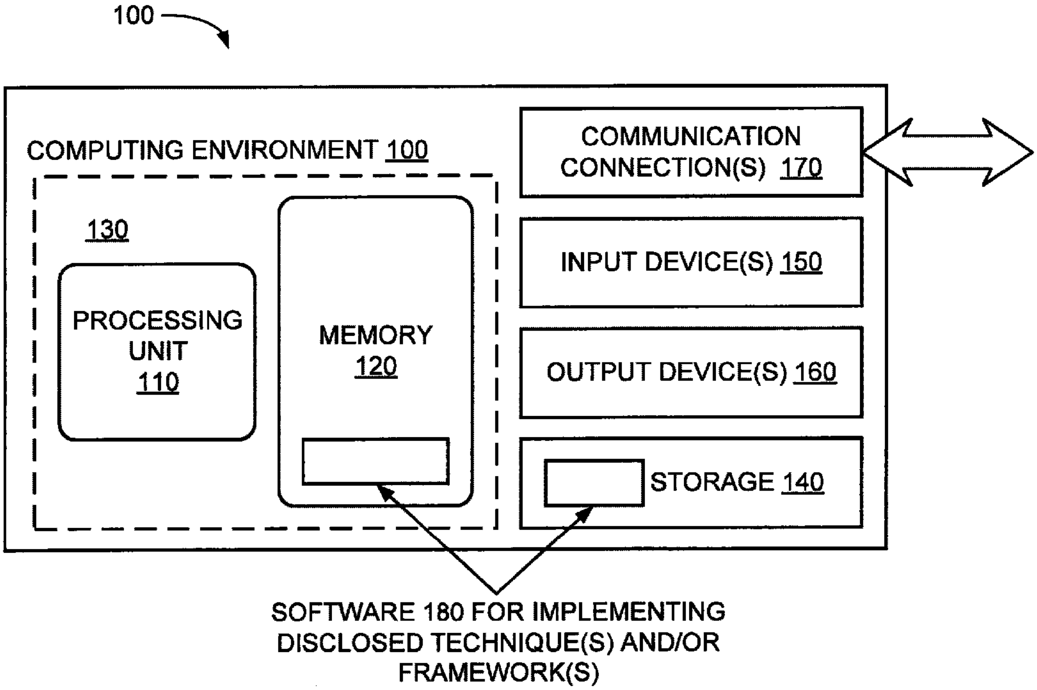
(51) **Cl.Int./Int.Cl.** ***G06Q 50/06*** (2012.01)

(72) **Inventeurs/Inventors:**
HAMMERSTROM, DONALD J., US;
MELTON, RON, US;
ESRAM, TRISHAN, US;
HATHAWAY, JOHN E., US

(73) **Propriétaire/Owner:**
BATTELLE MEMORIAL INSTITUTE, US

(74) **Agent:** SMART & BIGGAR LP

(54) **Titre : CONTROLE TRANSACTIF ET CADRE DE COORDINATION ET FONCTIONS DE BOITE A OUTILS ASSOCIEES**
(54) **Title: TRANSACTIVE CONTROL AND COORDINATION FRAMEWORK AND ASSOCIATED TOOLKIT FUNCTIONS**



(57) **Abrégé/Abstract:**
Disclosed herein are representative embodiments of methods, apparatus, and systems for facilitating operation and control of a resource distribution system (such as a power grid). For example, embodiments of the disclosed technology can be used to improve the resiliency of a power grid and to allow for improved consumption of renewable resources. Further, certain implementations facilitate a degree of decentralized operations not available elsewhere.

ABSTRACT OF THE DISCLOSURE

Disclosed herein are representative embodiments of methods, apparatus, and systems for facilitating operation and control of a resource distribution system (such as a power grid). For example, embodiments of the disclosed technology can be used to improve the resiliency of a power grid and to allow for improved consumption of renewable resources. Further, certain implementations facilitate a degree of decentralized operations not available elsewhere.

DEMANDE OU BREVET VOLUMINEUX

LA PRÉSENTE PARTIE DE CETTE DEMANDE OU CE BREVET COMPREND PLUS D'UN TOME.

CECI EST LE TOME 1 DE 2
CONTENANT LES PAGES 1 À 368

NOTE : Pour les tomes additionels, veuillez contacter le Bureau canadien des brevets

JUMBO APPLICATIONS/PATENTS

THIS SECTION OF THE APPLICATION/PATENT CONTAINS MORE THAN ONE VOLUME

THIS IS VOLUME 1 OF 2
CONTAINING PAGES 1 TO 368

NOTE: For additional volumes, please contact the Canadian Patent Office

NOM DU FICHIER / FILE NAME :

NOTE POUR LE TOME / VOLUME NOTE:

TRANSACTIVE CONTROL AND COORDINATION FRAMEWORK AND ASSOCIATED TOOLKIT FUNCTIONS

FIELD

This application relates generally to the field of power grid management and control.

SUMMARY

Disclosed below are representative embodiments of methods, apparatus, and systems for facilitating operation and control of a resource distribution system (such as a power grid). For example, embodiments of the disclosed technology can be used to improve the resiliency of a power grid and to allow for improved consumption of renewable resources. Further, certain implementations facilitate a degree of decentralized operations not available elsewhere.

"Transactive control and coordination" features market-like mechanisms for the selection of resources and demand-side assets in an electric power grid. The disclosed technology concerns new embodiments of transactive control and coordination. Such embodiments allow for transactive control and coordination where: (**1**) the system is implemented over large geographic areas; (**2**) the system is implemented across multiple grid regulation and/or business boundaries; (**3**) a large diversity of participating resources and loads are to be coordinated; and/or (**4**) the system desirably functions at multiple scales (*e.g.*, both large areas of the transmission region and at individual devices).

Locations on the electric power grid that perform one or more of the disclosed techniques of are sometimes referred to herein as "transactive nodes." Further, embodiments of the disclosed technology are described in terms of an "algorithmic framework," where the highest-level responsibilities that are to be conducted at a transactive node are discussed. In certain embodiments, two functional blocks within the algorithmic framework allow for the further incorporation of (**1**) "toolkit resource functions" and/or (**2**) "toolkit load functions." For example, depending on the unique features extant at a given transactive node (*e.g.*, certain types of generation resources, inelastic electrical loads, other loads that might be responsive to a price-like signal in a demand-responsive way), one or more toolkit functions and their unique functionality may be incorporated. These toolkit functions can respectively modify the formulation of the price-like signal by the framework, or modify the amount of load that is to be generated or consumed by assets at this grid location. The functions can also advise the control of responsive assets.

Embodiments of the disclosed technology can be used to realize the fully distributed coordination of electrical power grids. In certain embodiments, such coordination can be accomplished by having nearest circuit neighbors exchange *transactive signals*. Desirably, these signals include not only price and quantity signals for an imminent time interval, but also predicted signals for future time intervals. In certain implementations, at least two subclasses of *transactive signal* are used—one price-like and the other representing power. The *transactive signal* that represents power (the *TFS*) is usefully aggregated where the power is also combined in a circuit and represents the power flow between circuit neighbors; a price-like signal (the *TIS*) may fairly represent costs of multiple resources and incentives if such costs are proportionately added where the resources are injected into and where the incentives occur in the electrical circuit.

In certain implementations, and in contrast to system utilizing explicit bilateral markets, some of the disclosed systems use planned energy consumption as the feedback.

Also disclosed herein are tools and techniques for computing distributed relative power flow. For example, a distributed relative power flow method is formulated for electrical power systems. In certain embodiments, a node is allowed to allocate its generation or load changes among the power flows with its neighbors without the global knowledge of the power system. Further, in some embodiments, decisions are made independently at distributed locations to respond to incentive signals from distributed transactive control. The impacts of these decisions on power flow are desirably predicted, which is presently challenging to do with conventional power flow formulations. In certain embodiments, parallel computation is an inherent feature of the disclosed formulation.

Conventional power flow solvers, usually located at a central location, rely on the global knowledge of the power system to predict the impacts of generation or load changes on the power flow. However, it is challenging to predict the power flow by using such solvers at distributed locations, where only information from neighbor nodes may be available. This is not the case with embodiments of the disclosed distributed relative power flow formulations.

Embodiments of the disclosed power flow formulation can be used in a variety of environments. For example, such implementations can be used as part of a "smart grid" system, which heavily relies on two-way communication and transactive control.

Decisions to respond to incentive signals from transactive control cause power flow changes, which can be predicted in parallel at distributed locations, without knowledge of the entire power system.

Details of exemplary non-limiting embodiments of the disclosed technology are disclosed and illustrated in the sections below. Any one or more of the features, aspects, and/or functions described in any of the sections below or above can be used alone or in any combination or sub-combination with one another.

Embodiments of the disclosed methods can be performed using computing hardware, such as a computer processor or an integrated circuit. For example, embodiments of the disclosed methods can be performed by software stored on one or more non-transitory computer-readable media (*e.g.*, one or more optical media discs, volatile memory components (such as DRAM or SRAM), or nonvolatile memory or storage components (such as hard drives)). Such software can be executed on a single computer or on a networked computer (*e.g.*, via the Internet, a wide-area network, a local-area network, a client-server network, a cloud-based network, or other such network). Embodiments of the disclosed methods can also be performed by specialized computing hardware (*e.g.*, one or more application specific integrated circuits ("ASICs") or programmable logic devices (such as field programmable gate arrays ("FPGAs")) configured to perform any of the disclosed methods). Additionally, any intermediate or final result created or modified using any of the disclosed methods can be stored on a non-transitory storage medium (*e.g.*, one or more optical media discs, volatile memory or storage components (such as DRAM or SRAM), or nonvolatile memory or storage components (such as hard drives)) and are considered to be within the scope of this disclosure. Furthermore, any of the software embodiments (comprising, for example, computer-executable instructions which when executed by a computer cause the computer to perform any of the disclosed methods), intermediate results, or final results created or modified by the disclosed methods can be transmitted, received, or accessed through a suitable communication means.

The foregoing and other objects, features, and advantages of the invention will become more apparent from the following detailed description, which proceeds with reference to the accompanying figures.

According to one aspect of the present invention, there is provided a method for operating a transactive node in a market-based electrical-energy-allocation system, comprising: by computing hardware: computing incentive signal data, the incentive signal data comprising data indicative of a cost of electric energy at the transactive node at a current time interval and data indicative of a forecasted cost of electric energy at the transactive node at one or more future time intervals, wherein the incentive signal data further comprises data indicative of a confidence level that the data indicative of the cost of electric energy at the transactive node at the current time interval is accurate or data indicating a confidence level that the data indicative of the forecasted cost of electric energy at the transactive node at the one or more future time intervals is accurate; computing feedback signal data, the feedback signal data comprising data indicative of an electric load at the transactive node at the current time interval and data indicative of a forecasted load for electric energy at the transactive node at the one or more future time intervals; and transmitting the incentive signal data and the feedback signal data.

According to another aspect of the present invention, there is provided a method for operating a transactive node in a market-based electrical-energy-allocation system, comprising: by computing hardware: receiving incentive signal data at the transactive node from two or more neighboring transactive nodes, the incentive signal data from the two or more neighboring transactive nodes comprising data indicative of at least a cost of electric energy at a current time interval; computing aggregated incentive signal data based at least in part on the incentive signal data from the two or more neighboring transactive nodes; and transmitting the aggregated incentive signal data to a further transactive node; wherein the received incentive signal data further includes data indicating a confidence level of the received incentive signal data, or wherein the transmitted incentive signal data further includes data indicating a confidence level of the transmitted incentive signal data.

According to another aspect of the present invention, there is provided a method for operating a transactive node in a market-based electrical-energy-allocation system, comprising: by computing hardware: receiving feedback signal data at a transactive node from two or more neighboring transactive nodes, the feedback signal data from the two or more neighboring transactive nodes comprising data indicative of at least an electric load for electric energy at a current time interval; computing aggregated feedback signal data based at least in part on the feedback signal data from the two or more neighboring transactive nodes; and transmitting the aggregated feedback signal data to a further transactive node; wherein the received feedback signal data further includes data indicating a confidence level of the received feedback signal data, or wherein the transmitted feedback signal data further includes data indicating a confidence level of the transmitted feedback signal data.

According to another aspect of the present invention, there is provided a system for distributing electricity, comprising: a plurality of transactive nodes, each transactive node being associated with one or more electric resources, one or more electric loads, or a combination of one or more electric resources and one or more electric loads; and a network connected to the transactive nodes to facilitate communication between the transactive nodes, the transactive nodes being configured to exchange incentive and feedback signals with one another in order to determine an electrical demand in the system for a current time interval and to provide an electrical supply sufficient to meet the electrical demand for the current time interval; wherein, for each transactive node, the incentive signal comprises incentive signal data, the incentive signal data comprising data indicative of a cost of electric energy at the transactive node at a current time interval and data indicative of a forecasted cost of electric energy at the transactive node at one or more future time intervals, wherein the incentive signal data further comprises data indicative of a confidence level that the data indicative of the cost of electric energy at the transactive node at the current time interval is accurate or data indicating a confidence level that the data indicative of the forecasted cost of electric energy at the transactive node at the one or more future time intervals is accurate; wherein for each transactive node, the feedback signal comprises feedback signal data, the feedback signal data comprising data indicative of an electric load at the transactive node at the current time interval and data indicative of a forecasted load for electric energy at the transactive node at the one or more future time intervals.

According to another aspect of the present invention, there is provided a system for distributing electricity, comprising: a plurality of transactive nodes, each transactive node being associated with one or more electric resources, one or more electric loads, or a combination of one or more electric resources and one or more electric loads; and a network connected to the transactive nodes and facilitating communication between the transactive nodes, the transactive nodes being configured to exchange sets of signals with one another in order to determine an electrical demand in the system for a current time interval and to provide an electrical supply sufficient to meet the electrical demand for the current time interval, each set of signals including signals for determining the electric loads and supplies for the current time interval as well as signals for determining the electric loads and supplies for two or more future time intervals,

According to another aspect of the present invention, there is provided a system for distributing electricity, comprising: a plurality of transactive nodes, each transactive node being associated with one or more electric supplies, one or more electric loads, or a combination of one or more electric supplies and one or more electric loads; and a network connected to the transactive nodes and facilitating communication between the transactive nodes, the transactive nodes being configured to exchange sets of signals with one another in order to determine an electrical demand in the system for a current time interval and to provide an electrical supply sufficient to meet the electrical demand for the current time interval, a respective one of the transactive nodes being configured to compute its incentive and feedback signals using one or more functions selected from a library of functions.

According to another aspect of the present invention, there is provided a method, comprising: for a transactive node associated with one or more electric resources, one or more electric loads, or a combination of one or more electric reosources or one or more electric loads; selecting one or more functions from a library of functions, the selection being based at least in part on the type of one or more electric resources or electric loads associated with the transactive node; and configuring the transactive node to compute transactive signals using the selected one or more functions.

BRIEF DESCRIPTION OF THE DRAWINGS

The application contains at least one drawing executed in color. Copies of this patent application publication with color drawings will be provided by the Office upon request and payment of the necessary fee.

Figure 1 is a generalized example of a suitable computing hardware environment for a computing device with which several of the described embodiments can be implemented.

Figure 2 is a block diagram illustrating the transactive control concept.

Figure 3 is an illustration of the node-by-node changes to a transactive incentive signal as it flows from generation to end-use.

Figure 4 illustrates the dynamics of an electric vehicle charging example of the disclosed technology.

Figure 5 illustrates a simple topology for wind availability as will be used to illustrate an embodiment of the disclosed technology.

Figure 6 is a representation of toolkit functions for bulk power resources

Figure 7 is a graph showing the power generated at the transactive node represented by Figure **5** over time.

Figure 8 is a graph illustrating the unit costs of power for the current transactive control example.

Figure 9 is a graph that presents the hourly resource costs for wind power according to a conventional approach versus a transactive control approach.

Figure 10 is a graph that shows the cumulative cost comparison for a transactive control approach versus a conventional approach.

Figure 11 is a graph illustrating an example transactive incentive signal as it is affected by a wind power resource.

Figures **12A** and **12B** are a skeleton diagram of the algorithmic framework at a *transactive node*.

Figure 13 is a block diagram illustrating the example timing model.

Figure 14 is a diagram exemplifying the stacked component resource and incentive costs that compose a transactive signal.

Figure 15 is another diagram showing an example skeleton model of a standard *transactive node* that emphasizes the relationship between an exemplary overall methodology and the toolkit functions.

Figure 16 is a diagram illustrating one view of how multiscale intervals could be addressed by embodiments of the transactive system.

Figure 17 is a simple view of the responsibilities of a transactive node.

Figure 18 illustrates a basic transactive node model.

Figure 19 illustrates the constraint function transactive node component.

Figure 20 illustrates the load function transactive node component.

Figure 21 is a graph showing conceptual responses of methods to variation of an incentive signal.

Figure 22 illustrates a supply function node component.

Figure 23 illustrates a general transactive node.

Figure 24 is a flowchart illustrating an exemplary method for operating a transactive node according to certain embodiments of the disclosed technology.

Figure 25 is another flowchart illustrating an exemplary method for operating a transactive node according to certain embodiments of the disclosed technology.

Figure 26 is another flowchart illustrating an exemplary method for operating a transactive node according to certain embodiments of the disclosed technology.

Figure 27 is another flowchart illustrating an exemplary method for selecting a specific toolkit function from among a library of such toolkit functions.

Figure 28 illustrates the structure of numbered attributes at an exemplary transactive node.

Figure 29 shows an example state diagram for a transactive node.

Figure 30 is an exemplary connection state diagram that applies to transactive neighbors, system managers, assets, and local information.

Figure 31 is a diagram that illustrates TIS and TFS generation being decoupled.

Figure 32 is a diagram that illustrates TIS processing as may occur for some embodiments.

Figure 33 is a diagram illustrating an example where a perpetual exchange of signals might become sustained between two transactive node neighbors

Figure 34 is a graph showing weighting factors for a set of demonstration intervals (IST_0=**0**:**00**) using three different values of constant γ

Figures 35**A** and **35B** are a flowchart showing an example *toolkit framework* of functions and processes at a *transactive node*.

Figure 36 is a flowchart illustrating an exemplary "receive *transactive incentive signal*" process.

Figure 37 is a flowchart for an exemplary "calculate new *transactive signal* intervals" process.

Figure 38 is a flowchart illustrating an exemplary "formulate *TIS*" process.

Figure 39 is a flowchart of an exemplary "formulate TFS" process.

Figure 40 is a flowchart of an exemplary "sum total predicted load" process.

Figure 41 is a flowchart of an exemplary "calculate applicable toolkit load functions" process

Figure 42 is a flowchart of an exemplary "send *transactive signals*" process.

Figure 43 is a flowchart of an exemplary "calculate applicable *toolkit resource and incentive functions*" process.

Figure 44 is a flowchart of an exemplary "control *responsive asset systems*" process.

Figure 45 is a flowchart of an exemplary "sum total predicted resources" process.

Figure 46 is a flowchart of an exemplary "control responsive resource" process.

Figure 47 is a set of graphs showing predicted load $\hat{P}$ compared to measured load P for an example function.

Figure 48 is a set of graphs that show the linear least-squares error fit for each hour of the day, for day **4** given the measured data for an example function.

Figure 49 is a set of graphs that show the linear least-squares error fit for each hour of the day, for day **12** given the measured data for an example function.

Figure 50 is a set of graphs that show the linear least-squares error fit for each hour of the day, for day **14** given the measured data for an example function.

Figure 51 is a set of graphs showing predicted load $\hat{P}$ compared to measured load P for an example function.

Figure 52 is a set of graphs that show the linear least-squares error fit for each hour of the day, for day 4 given the measured data for an example function.

Figure 53 is a set of graphs that show the linear least-squares error fit for each hour of the day, for day 12 given the measured data for an example function.

Figure 54 is a set of graphs that show the linear least-squares error fit for each hour of the day, for day 14 given the measured data for an example function.

Figure 55 is a graph of power vs. wind speed for wind turbines for an example function.

Figure 56 is a graph of a hypothetical supply stack.

Figure 57 is a diagram showing a sample daily DowJones Mid-C hourly index.

Figure 58 is a plot of examplary overall cost of energy for hydropower for each season for an example function.

Figure 59 shows example graphs for $DIST(T/S_0)$ and $\Phi(T/S_0)$.

Figure 60 is a graph showing a typical water heater power consumption during week and weekend days.

Figure 61 is an example profile of $P_S(t)$.

Figure 62 is a plot of a winter profile of $T_{OSP}(t)$ that uses the winter parameters.

Figure 63 is a plot of a summer profile of $T_{OSP}(t)$ that uses the summer parameters.

Figure 64 is a graph of the predicted electrical power consumption for 1000 thermostatically controlled residential buildings where $T_o = 10°C$

Figure 65 is a graph of the predicted electrical power consumption for 1000 thermostatically controlled residential buildings where $T_o = 0°C$

Figure 66 is a graph of the predicted electrical power consumption for 1000 thermostatically controlled residential buildings where $T_o = 0°C$; $\Delta T_{DRSP} = -2°C$ from 8:00 to 10:00 am.

Figure 67 is a graph of the predicted electrical power consumption for 1000 thermostatically controlled residential buildings where $T_o = 0°C$; $K_{DRP} = 0.75$ from 8:00 to 10:00 am.

Figure 68 is a plot showing results of simulating MATLAB code with one response level.

Figure 69 is another plot showing results of simulating MATLAB code with one response level.

Figure 70 is another plot showing results of simulating MATLAB code with one response level.

Figure 71 is another plot showing results of simulating MATLAB code with one response level.

Figure 72 is another plot showing results of simulating MATLAB code with one response level.

Figure 73 is a plot showing results of simulating MATLAB code with two response levels.

Figure 74 is another plot showing results of simulating MATLAB code with two response levels.

Figure 75 is another plot showing results of simulating MATLAB code with two response levels.

Figure 76 is another plot showing results of simulating MATLAB code with two response levels.

Figure 77 is another plot showing results of simulating MATLAB code with two response levels.

Figure 78 is an example plot of a lighting load.

Figure 79 is an example plot of a refrigerator load.

Figure 80 is an example plot of a cooking range load.

Figure 81 is an example plot of a dishwasher load.

Figure 82 is an example plot of a clothes washer load.

Figure 83 is an example plot of a clothes dryer load.

Figure 84 is an example plot of a miscellaneous electric load.

Figure 85 is a block diagram showing an example model of ramp up and ramp down periods.

Figure 86 is a block diagram of an example block input/output function model.

Figure 87 is a set of plots for $DIST(TIS_0)$ and $\Phi(b)$.

Figure 88 is an illustration of TOU voltage control concurrent with shedding water heaters.

Figure 89 is a series of plots that show possible scenarios for changes in generation during one interval.

Figure 90 is an infrastructure cost control diagram.

Figure 91 shows a graph illustrating the improvement of uninitialized infrastructure cost estimate for different α parameter selections assuming **5**-minute update intervals.

Figure 92 shows a graph illustrating the uninitialized correction of TIS over time for different α parameter selections assuming **5**-minute update intervals.

Figure 93 is a diagram of an exemplary block input/output function model.

Figure 94 is a graph illustrationg an example for one iteration at a given time.

Figure 95 is a diagram that shows the specified strategy during a month.

Figure 96 is a graph illustrating power operations concepts.

Figure 97 is a diagram of an exemplary block input/output function model.

Figure **98** is a first diagram illustrating an example power flow computation.

Figure **99** is a second diagram illustrating an example power flow computation.

Figure **100** is a third diagram illustrating an example power flow computation.

DETAILED DESCRIPTION

1 General Considerations

Disclosed below are representative embodiments of methods, apparatus, and systems for facilitating operation and control of a resource distribution system (such as a power grid). The disclosed methods, apparatus, and systems should not be construed as limiting in any way. Instead, the present disclosure is directed toward all novel and nonobvious features and aspects of the various disclosed embodiments, alone and in various combinations and subcombinations with one another. Furthermore, any one or more features or aspects of the disclosed embodiments can be used in various combinations and subcombinations with one another. The disclosed methods, apparatus, and systems are not limited to any specific aspect or feature or combination thereof, nor do the disclosed embodiments require that any one or more specific advantages be present or problems be solved.

Although the operations of some of the disclosed methods are described in a particular, sequential order for convenient presentation, it should be understood that this manner of description encompasses rearrangement, unless a particular ordering is required by specific language set forth below. For example, operations described sequentially may in some cases be rearranged or performed concurrently. Moreover, for the sake of simplicity, the attached figures may not show the various ways in which the disclosed methods can be used in conjunction with other methods. Additionally, the description sometimes uses terms like “determine” and “generate” to describe the disclosed methods. These terms are high-level abstractions of the actual operations that are performed. The actual operations that correspond to these terms may vary depending on the particular implementation and are readily discernible by one of ordinary skill in the art. Furthermore, as used herein, the term “and/or” means any one item or combination of items in the phrase.

Any of the disclosed methods can be implemented using computer-executable instructions stored on one or more computer-readable media (*e.g.*, non-transitory computer-readable media, such as one or more optical media discs, volatile memory components (such as DRAM or SRAM), or nonvolatile memory components (such as hard drives)) and executed by a processor in a computing device (*e.g.*, a computer, such as any commercially available computer). Any of the computer-executable instructions for implementing the disclosed techniques as well as any intermediate or final data created and used during implementation of the disclosed systems can be stored on one or more computer-readable media (*e.g.*, non-transitory computer-readable media). The computer-executable instructions can be part of, for example, a dedicated software application or as part of a software agent's transport payload that is accessed or downloaded via a network (*e.g.*, a local-area network, a wide-area network, a client-server network, or other such network).

Such software can be executed on a single computer (*e.g.*, a computer embedded in or electrically coupled to a sensor, controller, or other device in the power grid) or in a network environment. For example, the software can be executed by a computer embedded in or communicatively coupled to a sensor for measuring electrical parameters of a power line or electrical device, a synchrophasor sensor, a smart meter, a control unit for a home or household appliance or system (*e.g.*, an air-conditioning unit; heating unit; heating, ventilation, and air conditioning ("HVAC") system; hot water heater; refrigerator; dish washer; washing machine; dryer; oven; microwave oven; pump; home lighting system; electrical charger; electric vehicle charger; home electrical system; or any other electrical system having variable performance states), a control unit for a distributed generator (*e.g.*, photovoltaic arrays, wind turbines, or electric battery charging systems), a control unit for controlling the distribution or generation of power along the power grid (*e.g.*, a transformer, switch, circuit breaker, generator, resource provider, or any other device on the power grid configured to perform a control action), and the like. Further, any of the control units can also include or receive information from one or more sensors. Any of the transactive nodes described herein can be formed by such sensors, meters, control units, and/or other such units.

For clarity, only certain selected aspects of the software-based embodiments are described. Other details that are well known in the art are omitted. For example, it should be understood that the software-based embodiments are not limited to any specific computer language or program. For instance, embodiments of the disclosed technology can be implemented by software written in C++, Java, Perl, JavaScript, Adobe Flash, Python, JINI, .NET, Lua or any other suitable programming language. Likewise, embodiments of the disclosed technology are not limited to any particular computer or type of hardware. Details of suitable computers and hardware are well known and need not be set forth in detail in this disclosure. Furthermore, any of the software-based embodiments (comprising, for example, computer-executable instructions which when executed by a computer cause the computer to perform any of the disclosed methods) can be uploaded, downloaded, or remotely accessed through a suitable communication means. Such suitable communication means include, for example, the Internet, the World Wide Web, an intranet, software applications, cable (including fiber optic cable), magnetic communications, electromagnetic communications (including RF, microwave, and infrared communications), electronic communications, or other such communication means.

The disclosed methods can also be implemented by specialized computing hardware that is configured to perform any of the disclosed methods. For example, the disclosed methods can be implemented by a computing device comprising an integrated circuit (*e.g.*, an application specific integrated circuit ("ASIC") or programmable logic device ("PLD"), such as a field programmable gate array ("FPGA")). The integrated circuit or specialized computing hardware can be embedded in or directly coupled to a sensor, control unit, or other device in the power grid. For example, the integrated circuit can be embedded in or otherwise coupled to a synchrophasor sensor, smart meter, control unit for a home or household appliance or system, a control unit for a distributed generator, a control unit for controlling power distribution on the grid, or other such device.

Figure **1** illustrates a generalized example of a suitable computing hardware environment **100** for a computing device with which several of the described embodiments can be implemented. For example, any of the transactive nodes disclosed herein can be implemented by a computing hardware environment, such computing environment **100**. The computing environment **100** is not intended to suggest any limitation as to the scope of use or functionality of the disclosed technology, as the techniques and tools described herein can be implemented in diverse general-purpose or special-purpose environments that have computing hardware.

With reference to Figure **1**, the computing environment **100** includes at least one processing unit **110** and memory **120**. In Figure **1**, this most basic configuration **130** is included within a dashed line. The processing unit **110** executes computer-executable instructions. In a multi-processing system, multiple processing units execute computer-executable instructions to increase processing power. The memory **120** may be volatile memory (*e.g.*, registers, cache, RAM), non-volatile memory (*e.g.*, ROM, EEPROM, flash memory), or some combination of the two. The memory **120** stores software **180** for implementing one or more of the described techniques for operating or using the disclosed systems. For example, the memory **120** can store software **180** for implementing any of the disclosed techniques.

The computing environment can have additional features. For example, the computing environment **100** includes storage **140**, one or more input devices **150**, one or more output devices **160**, and one or more communication connections **170**. An interconnection mechanism (not shown) such as a bus, controller, or network interconnects the components of the computing environment **100**. Typically, operating system software (not shown) provides an operating environment for other software executing in the computing environment **100**, and coordinates activities of the components of the computing environment **100**.

The storage **140** can be removable or non-removable, and includes magnetic disks, magnetic tapes or cassettes, CD-ROMs, DVDs, or any other tangible storage medium which can be used to store information in a non-transitory manner and which can be accessed within the computing environment **100**. The storage **140** can also store instructions for the software **180** implementing any of the described techniques, systems, or environments. The input device(s) **150** can be a touch input device such as a keyboard, mouse, touch screen, pen, or trackball, a voice input device, a scanning device, or another device that provides input to the computing environment **100**. The output device(s) **160** can be a display, touch screen, printer, speaker, or another device that provides output from the computing environment **100**.

The communication connection(s) **170** enable communication over a communication medium to another computing entity. The communication medium conveys information such as computer-executable instructions, an agent transport payload, or other data in a modulated data signal. A modulated data signal is a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media include wired or wireless techniques implemented with an electrical, optical, RF, infrared, acoustic, or other carrier.

The various methods, systems, and interfaces disclosed herein can be described in the general context of computer-executable instructions stored on one or more computer-readable media. Computer-readable media are any available media that can be accessed within or by a computing environment but do not encompass transitory signals or carrier waves. By way of example, and not limitation, with the computing environment **100**, computer-readable media include tangible non-transitory computer-readable media, such as memory **120** and storage **140**.

The various methods, systems, and interfaces disclosed herein can also be described in the general context of computer-executable instructions, such as those included in program modules, being executed in a computing environment on a target processor. Generally, program modules include routines, programs, libraries, objects, classes, components, data structures, and the like that perform particular tasks or implement particular abstract data types. The functionality of the program modules may be combined or split between program modules as desired in various embodiments.

Computer-executable instructions for program modules may be executed within a local or distributed computing environment. As noted, the disclosed technology is implemented at least part using a network of computing devices (*e.g.*, any of the computing device examples described above). The network can be implemented at least in part as a Local Area Network ("LAN") using wired networking (*e.g.*, the Ethernet IEEE standard **802.3** or other appropriate standard) or wireless networking (*e.g.* one of the IEEE standards **802.11**a, **802.11**b, **802.11**g, or **802.11**n or other appropriate standard). Furthermore, at least part of the network can be the Internet or a similar public network.

1.1 Acronyms and Abbreviations

This disclosure sometimes makes reference to the following acronyms:

HVAC	heating, ventilating and air conditioning
IST	interval start time
LMP	locational marginal price
RMS	root mean square
TCS	transactive coordination system
TFS	transactive feedback signal
TIS	transactive incentive signal
UTC	Coordinated Universal Time

1.2 Terms

This disclosure will sometimes make reference to the following terms, whose non-limiting definitions are provided below. These definitions do not necessarily apply in all instances and may vary depending on the context.

advisory control signal	A signal that is transmitted by a *transactive node* to its local responsive asset systems advising these systems to change their energy consumption or generation
asset model	A usually dynamic model of an asset system (e.g., a population of electric water heaters) that can predict its change in load or change in supply in light of an event (e.g., a curtailment of the asset system).
locational marginal price	A unit price of energy that represents the spatial and temporal price of the marginal supply resource. Today, *locational marginal price* is calculated centrally.
non-transactive energy	Refers to energy that can be exchanged between *transactive nodes* of a *transactive coordination system* and entities that reside outside the boundaries of the *transactive coordination system*
relaxation criterion	A criterion against which changes in subsequent *transactive signal*s are compared. If changes are significant based on this criterion, then new *transactive signal*s are calculated and published.
transactive coordination system	A distributed system-of-systems in which *transactive nodes* coordinate the balance between energy resources and loads by communicating *transactive signal*s
toolkit function	A function that is invoked by the *transactive node object model* to represent the unique set of incentives, resources, and loads that are managed at the *transactive node*. Includes two subclasses—*toolkit resource and incentive functions* and *toolkit load functions*.
toolkit load	One type of a plurality of *toolkit functions* that calculates load, change in

function	elastic load, and control signals for the specific demand-side assets at a *transactive node*
toolkit resource and incentive function	One type of a plurality of *toolkit functions* that calculates incentive costs, supply energy, and energy costs for the specific incentives and supply resources at a *transactive node*. Includes *toolkit resource functions* and *toolkit incentive functions*.
transactive energy	Energy that is exchanged between *transactive nodes* of a *transactive coordination system*
transactive feedback signal	One of a plurality of subclasses of *transactive signals*. Represents predicted aggregate power flow between two neighboring *transactive nodes*.
transactive incentive signal	One of a plurality of subclasses of *transactive signal*. Represents the delivered unit cost of energy at a system location.
transactive neighbors	Adjacent *transactive nodes* that exchange energy and are therefore obligated to exchange *transactive signals* with one another. This term may be equivalently stated as *neighboring transactive nodes* or *circuit neighbors*.
transactive node	A node that participates in a *transactive coordination system* to send and receive *transactive signals*
transactive node object model	The formal state model that resides at a *transactive node* and defines its behaviors, interactions, and interfaces. This term usually refers to the common responsibilities of *transactive nodes* that are interoperable, standardized.
transactive signal	A class of signal shared between *transactive neighbors*

2 Introduction

This section introduces some of the basic concepts of the disclosed transactive control and coordination technology. Figure **2** is a block diagram illustrating a general system **200** for implementing transactive control. The figure represents a simple electric power system topology **200** with power flowing from generation resources on the right through the components of the system to loads on the left.

At any point in the topology where one can affect the flow of power, operational objectives may be taken into account. In the transactive control technique of the disclosed technology, these objectives can be monetized and included in a signal referred to as the "transactive incentive signal" (TIS). If at a given point, one should reduce load below that point, then the monetization computations will result in altering (e.g., raising) the value of the TIS. If, on the other hand, it is beneficial to add load below that point, then the computations will alter (e.g., lower) the value of the TIS in the opposite direction. In other words, by using embodiments of the disclosed transactive system, one can represent operational objectives to responsive elements of the system and incentivize them to change their behavior in response to the monetized objectives. In Figure **2**, this is represented by the arrow from right-to-left labeled "operational objectives."

The responsive elements of the system also play an active role through making information available about their planned consumption of electric power. This is represented by the arrow from left-to-right labeled "status and opportunities." In embodiments of the disclosed technology, information about the future forecast of the plans for generation resources and constraints associated with the flow of power through the system interact with temporally aligned information about the planned behavior or loads or other responsive resources. Local storage systems are an example of another type of responsive resource that may be thought of as being a positive, neutral (not consuming), or negative load.

With this general background, the following additional features of the transactive control and coordination system will now be introduced.

- Transactive Control: A single, integrated, smart grid incentive signaling approach utilizing an economic signal as the primary basis for communicating the desire to change the operational state of responsive assets.

- Transactive Incentive Signal (TIS): A representation of the actual delivered cost of electric energy at a specific system location (e.g., at a transactive node). Includes both the current value and a forecast of future values. In certain embodiments, the current incentive signal value refers to the value for the imminent (or next-to-occur) interval.
- Transactive Feedback Signal (TFS): A representation of the net electric load at a specific system location (e.g., between neighboring transactive nodes). Includes both the current value and a forecast of future values. In certain embodiments, the current value refers to the feedback signal value for the imminent (or next-to-occur) interval.

2.1 What is a transactive control node?

The basic operational unit of embodiments of the illustrated transactive control technique is the transactive control node. In certain implementations, the transactive control node responds to system conditions as represented by incoming Transactive Incentive Signals and Transactive Feedback Signals through (a) incorporation of local asset status and other local information; (b) decisions about behavior of local assets; and/or (c) updating both transactive incentive and feedback signals. Inputs are used by the node to compute incentive and feedback signals. Further, in some embodiments, each signal is a sequence of forecasts for a time-series, so inputs will also be sequences of future (forecast/planned) values

Transactive control nodes may be implemented any place in the power system topology, preferably where it is possible to affect the flow of power in the system. This is true in both the bulk power system and carries through into the distribution system down to the end-use level. For example, embodiments of the disclosed technology can be used in a large region of the power grid (e.g., a large interconnected region of the transmission grid, sometimes referred to as a transmission zone), a distribution utility service territory, or for any other sized region, area, or space (e.g., at the substation level, at the feeder level, at a building level, or even at the household level. Transactive control nodes may be implemented down to the level of individual devices. One may also implement transactive control nodes that manage a collection of devices as an aggregated responsive asset or asset system.

2.2 An end-to-end view

Figure **3** is an illustration **300** of the node-by-node changes to a transactive incentive signal (TIS) as it flows from generation to end-use. In particular, Figure **3** provides a high level end-to-end view of the flow of transactive incentive signals through a transactive control and coordination system. In the figure, the TIS begins at a generation resource with the TIS values representing the generation cost. To simplify the example, transmission costs are also included so that when the signal is received at the utility-level, it represents the full cost of power delivered to the utility.

At the utility level, the utility has the opportunity to introduce local information and operational objectives. For example, the utility may wish to avoid demand charges associated with peak loads. The financial impact of peak loads can be used in calculating TIS values to incentivize load shifting.

In the example, there are also renewable generation assets local to the utility. The utility may also incentivize consumption of energy from these assets through the TIS. On the right hand side of Figure **3**, one can see the TIS presented to responsive assets as an aggregation of the costs to delivery power to the end-uses including generation costs, constraints, and operational objectives.

Missing from this example is the transactive feedback signal representing the behavior of the responsive assets. A feature of certain embodiments of the transactive control technique is that this signal and the transactive incentive signal are both used at a transactive control node to make decisions about the behavior of responsive assets controlled at that node or to be incentivized by that node. This interaction between the TIS and TFS takes place based on the forecast of cost of power delivered and the behavior of responsive assets. Through this interaction, a form of closed loop control is achieved. The decision logic and algorithmic functions of the transactive control node are desirably constructed in such a manner as to have convergence and to avoid oscillation.

2.3 An end-to-end view via an illustrative example

One can better understand this interaction between the TIS and TFS through a simple qualitative example. Consider the following scenario. On a distribution feeder, imagine a pole top transformer feeding three houses. Each home has an electric vehicle. For this example, assume that each of the vehicle owners will want to fast charge their vehicle. With the normal base load for the three houses, all three vehicles fast charging will overload the pole top transformer.

In this example, the pole top transformer is receiving a TIS from upstream (presumably from the substation) and a TFS from each of the houses. The TFS from each house includes information about the planned charging activity for the corresponding electric vehicle. The transformer desirably makes decisions about whether to change the value of the TIS based on the current and future load as represented by aggregating the TFS from each house. It also may take into account other information, such as the ambient air temperature, weather forecasts, operating history, and so forth.

The three electric vehicles in this example, EV**1**, EV**2**, and EV**3**, each have different charging strategies. EV**1** is capable of flexible charging, meaning that the rate of charge can be varied. EV**2** charges at any cost. EV**3** is a bargain hunter and will schedule charging when cost is low.

For this example, assume the following: EV**1** desires to charge at **5**PM, EV**2** wishes to charge at **6**PM and EV**3** wishes to charge at **7**PM. Assume as well that there is a typical diurnal load curve for the three houses seen in this example as the combined load at the transformer. The pole-top transformer has a load rating of **40** kW. As long as the load is below **40** kW, the service life of the transformer is not being degraded. If the load is above **40** kW, then the service life of the transformer is reduced depending on factors including the load, the duration of load above the **40**-kW limit, ambient air temperature and possibly other factors. The operating principle for the transformer's update to the TIS is a computation in which the monetary impact of load is computed based on the forecasted duration above the limit and the other factors mentioned. This computation can be performed with information about the cost to replace the transformer, the rated service life, and if desired, economic factors such as the cost of money. The point is that the impact of overloading the transformer is monetized and the result used to change the forecast value of the TIS.

The electric vehicle smart chargers may then respond to the change in TIS value (e.g., increased for overloading) and adjust their plans accordingly. A back and forth exchange, a negotiation if you will, takes place through the exchange of TIS and TFS updates. When the negotiation settles, then the "agreed" solution to consumption should be stable barring other perturbations.

A key challenge in this negotiation is to avoid oscillation. The algorithms and decision logic for both the smart charger and the transformer desirably have appropriate damping factors to drive the negotiation to a stable, non-oscillatory result. In this simple example, a qualitative result is presented to illustrate the nature of the interaction.

Figure 4 illustrates the dynamics of the electric vehicle charging example. As described above, EV1 forecasts that it will start charging at 5pm (hour 17), EV2 forecasts that it will start charging at 6pm (hour 18) and EV3 forecasts that it will start charging at 7pm (hour 19). None of the EV smart chargers have knowledge of the plans of the other. Information is communicated via their forecasts sent to the pole-top transformer and the resulting changes in the forecast TIS value.

In the figure, the broad dashed line represents the forecast total load. Notice that between hours 16 and 17, it simply tracks the normal diurnal load pattern. When the charging plans of the EV's are revealed through the TFS sent to the pole-top tranformer's transactive control node the forecast total load remains below the transformer's load limit until the time that EV2 proposes to start charging. Note that, in this example, all vehicles are proposing a level-2 fast charge initially.

When EV2 proposes to begin charging at hour 18, the forecast total load goes above the load limit. The TIS correspondingly increases above the TIS that is associated with the normal diurnal load. EV3's proposal to begin charging at hour 19 pushes the forecast load even higher. If all three vehicles are level-2 charging, the load approaches 10 kW above the load limit. With the three proposed charging times revealed, the TIS is adjusted and the vehicles respond. For this example, the result is simplified by showing the final result. In practice several iterations would typically be used to achieve the final, stable result.

The final result, as illustrated in Figure 4, shows that EV1 adapts its plans based on its flexible charging strategy. EV2 does not modify its plan. Remember this is the vehicle that will charge at any price. EV3, the bargain hunter, chooses to shift charging to a night time hour when prices are even lower than its original proposal to begin charging at hour 19. As seen in the figure, EV1's flexible charging strategy offsets EV2's charge at any price to maintain the total load just at the transformer's load limit.

This simple example illustrated the basic principle of the transactive control technique. The technique can be applied at any point in the power systemand can coordinate monetized energy impacts and the behaviors of responsive loads where such devices and opportunities exist. Consider, for example, a battery storage system at a distribution substation. The associated transactive control node would be making decisions about whether to charge, discharge, or do nothing with the battery system based on the incoming TISs, the incoming TFSs, local conditions such as the state of the battery system, and updating the TIS and TFS it sends to neighboring transactive control nodes accordingly. Transactive control nodes can be deployed throughout the power system from generation resources, through the transmission system, and in the distribution system down to end uses. The technique can be applied within end use points including residential, commercial and industrial uses to manage the behavior of responsive systems and devices.

2.4 Extended Example

The example above showed the use of the transactive control technique at end-use points within a distribution system. In this section, a further example of the transactive control and coordination system is considered. This example further illustrates the use of the technique to use local responsive assets to help facilitate the integration of intermittent renewable energy resources.

In order to facilitate discussion of this example, first consider the formalization of the transactive control technique. This allows the use of standard way of referring to the functional elements of an implemented transactive control and coordination system.

For embodiments of the disclosed technology, consider a formal model of the functionality of transactive control nodes. A transactive control node object state model has been defined and is the basis for implementing a transactive node object model (TNOM). This approach is scalable, algorithmic and supports explicit consideration of interoperability through the formal specification of both the syntax and semantics of the transactive incentive signal and transactive feedback signal. The "responsibilities" of a transactive control node summarized earlier are formally represented in the object model.

For embodiments of the disclosed technology, a standardized approach to implementation is made possible through the design and implementation of a "toolkit." The toolkit includes well-defined interfaces to utility responsive asset systems and simple, common algorithms for updating transactive signals and determining "control" signals to responsive asset systems.

In designing the toolkit, functions for resources and loads can be defined. The resource functions are primarily defined for the bulk power system and represent systems that supply power. At the utility level, functions associated with local resources or utility concerns such as avoiding demand charges are defined. Load functions can be defined that are associated with the different classes of loads or with local resources such as battery storage systems that may have load or resource behaviors (which are treated as negative loads.)

In embodiments of the disclosed technology, the resource functions include functions from a wide variety of categories. For example, in certain embodiments, the resource functions include one or more of:

1. **Imported electrical energy**
    1.1. Non-transactive imported energy
    1.2. Transactive imported energy
2. **Renewable energy resource**
    2.1. Wind energy
    2.2. Solar energy
    2.3. Hydropower
3. **Thermal generation**
4. **General infrastructure cost**
5. **System constraints**
    5.1. Transmission constraints
    5.2. Equipment and line constraints
6. **System energy losses**
    6.1. Transmission losses
    6.2. Distribution losses
    6.3. Device/component losses
7. **Demand charges**
8. **Market impacts**

In embodiments of the disclosed technology, the load functions include one or more functions from the following categories: (**1**) inelastic, (**2**) elastic with limited numbers of discrete events available, (**3**) elastic with daily events available, or (**4**) elastic with a continuum or near continuum of responses available. There can then exist a matrix of these four categories, with specific loads that fit into one or more of these categories. For example purposes only, the following is a list of example load functions that should not be construed as limiting in any manner. For instance, load functions can be created for a wide variety of assets or asset systems that that can be used in embodiments of the disclosed technology (e.g., for a residence, there may be functions for a variety of different assets and/or asset systems, such as responsive water heaters, thermostats, clothes dryers, web portals, in-home displays, or other such assets and asset systems).

1. **Bulk inelastic load**
    - 1.1. Bulk commercial load
    - 1.2. Bulk industrial load
    - 1.3. Bulk residential load
    - 1.4. Small wind generator negative load
    - 1.5. small-scale distributed generator negative load
    - 1.6. Small-scale solar generator negative load
2. **General event-driven demand response**
    - 2.1. Commercial
    - 2.2. Distribution system voltage control
    - 2.3. Residential behavior
        - 2.3.1. Portals
3. **General time-of-use demand response**
    - 3.1. Battery storage
    - 3.2. Commercial
    - 3.3. Residential behavioral
        - 3.3.1. Portals
    - 3.4. Residential
    - 3.5. Distribution system voltage control
4. **General real-time continuum demand response**
    - 4.1. Battery storage
    - 4.2. Commercial
    - 4.3. Residential behavioral
        - 4.3.1. Portals
    - 4.4. Residential

It should be understood that in embodiments of the disclosed technology, a transactive node may host multiple toolkit funcitons, including any combination of multiple resource and incentive functions, multiple load functions, or combinations of both resource and incentive and load functions. For instance, the resource and/or incentive functions used at a transactive node will typically depend on the location of the transactive node in a power grid topology, and on the one or more resources and/or loads for which the transactive node is responsible. This ability to "mix and match" resource and incentive functions while still maintaining a common transactive signal communication structure gives embodiments of the disclosed technology wide flexibility and scalability for implementing a transactive control system.

2.4.1 An example using wind resources

For this example, consider the following general conditions and objectives: (a) the predicted transactive incentive signal increases when wind energy decreases and visa versa; (b) the transactive incentive signal is communicated and mixed between transactive nodes; and/or (c) assets respond to improve consumption of wind when wind energy is available or near where wind is available.

For purposes of this example, also consider the simple topology **500** illustrated in Figure **5**. In the left hand side of the figure, a transactive control node can be observed with two generation resources. The lower illustration in the figure represents conventional generation such as a coal fired power plant. The upper illustration represents wind turbines. On the right hand side of the figure, one can see a transactive control node with three types of assets: conventional resistive load in the form of a water heater, a distributed energy resource in the form of battery storage, and a distribution system voltage control system represented by the cartoon with wires and power poles. For this example, the two transactive control nodes are communicating with each other through the exchange of transactive incentive signals and transactive feedback signals. Note that the transactive control node on the left is associated with features of the bulk power system – bulk generation resources – while the transactive control and coordination system on the right is associated with assets in the distribution system.

Consider now the toolkit load functions associated with the resources shown in the left hand side of illustration **600** in Figure **6**, which shows representations of toolkit functions for bulk power resources. A graphical representation of these toolkit functions is also shown in Figure **6**.

For conventional generation, toolkit resource function **#2** shown in Figure **6**, the function is a single point representing a fixed cost of production. The vertical access represents cost in $ / MWh and the horizontal axis the power produced. For purposes of this example, assume that this resource operates at a fixed point ignoring for this example ramping and any other factors that would cause the power output to vary.

The other example of wind power, toolkit resource function **#1**, is more complicated. In this case, assume a cost of power that is inversely proportional to the power output of the system. Thus, when there is low wind and low production the cost per unit of power is high. On the other hand, when there is high wind and corresponding high power output the cost is low. It should be noted that there are many possible ways to construct the resource functions. The underlying question is how to assign cost – to monetize the activity of the resource asset. In embodiments of the disclosed technology, one should assign cost in a way that incentivizes desired outcomes. In this example, the resource function defined for the wind resource has lowest cost when there is an abundance of wind power thus incentivizing consumption of wind power when it is available. Another consideration when evaluating potential resource functions is that candidate resource functions for a given asset should ensure the same total cost over relatively long periods of time.

Having defined resource functions allows one to look at their behavior over time. Figure **7** is a graph **700** that depicts the power generated at transactive node **#1**. Base generation is shown as constant at **10**MW. Wind generation varies from **10**MW for the first **30** hours dropping to zero (**0**) thereafter.

With this forecast of power production in mind, consider the forecast of cost of power from these two resources both with current approaches and with the transactive control approach using embodiments of the resources functions disclosed herein.

In this example, short-term power trading on spot or even day-ahead markets is ignored. In this case, the cost of power will be an aggregated value based on the fixed rate associated with each of the two resources. From the point of view of today's consumer, the cost of power is at a fixed rate – thus there is no incentive to change consumption behavior associated with the cost of power.

Figure **8** is a graph **800** illustrating the unit costs of power for the current transactive control example. In this case, the base generation is still provided at a fixed cost as previously shown. The unit cost of wind power is at a relatively lower cost while the wind is blowing and rises when the wind dies – eventually becoming infinite when wind power is unavailable. The aggregate cost, that seen by consumers, is an average (possibly weighted) of the two representing the incentive to consume when wind is available at a cost below normal base generation cost and to not consume when wind is unavailable at a cost above normal base generation cost.

Embodiments of the disclosed technology provide a scheme that incentivizes the desired behavior – preferentially to consume wind power. But what about the long-term cost objective? Let us compare how costs accumulate over time. Figure **9** is a graph **900** that presents a comparison of hourly resource costs with or without transactive control. Given the examples, the resource function for base generation, the hour cost for that resource is the same in either case. For the wind resource, however, the hourly cost is quite different. As the system and economics are currently formulated, the wind resource is only compensated when it is producing power. The rate is fixed and costs should be recovered based on an estimate over the long term of the percentage of time the resource will be available. This is represented by the line in Figure **9** that starts at the hourly cost of **100** and then drops to zero (**0**) at hour **31**. In contrast, the hourly cost for the wind resource is constant at **40** using transactive control. This is because during the period of time when wind power is available, loads are incentivized to consume via a lower cost (e.g., using the transactive incentive signal) and incentivized to not consume via a higher cost when wind is not available. The cost of wind production still should be recovered. So over the excess cost recovery when wind is not available (as compared to base generation cost) is used to make the wind producer whole resulting in an apparent fixed hourly cost.

Integrating the hourly costs allows one to check the long-term criteria – that costs should be the same over the long term for transactive versus the non-transactive approaches. Figure **10** is a graph **1000** that shows this cumulative cost comparison and shows that the transactive control technique can be formulated in such a manner as to achieve this objective.

Now that the formulation of toolkit resource functions have been considered, example differences between conventional approaches and embodiments of the transactive approach can be summarized. For instance, the resource functions for generation assets of the disclosed technology create a transactive incentive signal as depicted in graph **1100** of Figure **11**. The dynamics of the signal are as described above in the discussion of unit costs.

Attention can be shifted to the consumption, or load, side of the computation. From a behavioral or responsiveness point of view, loads will be mixed. Some will be controllable;

in other words, the loads will have the potential to respond to an incentive signal. Still further, in some instances, some loads will also be capable of acting as a load or a generation resource. For example, a battery system may have either behavior, and decisions about the battery may be made about when to charge, discharge, and/or at what rates. In this respect, a battery load may be highly responsive. For any given class of load assets, one may construct one or more load toolkit functions. These functions desirably take into account the load functions for other distribution system assets, and are discussed in more detail below.

Embodiments of the disclosed technology implement a distributed system for engaging responsive assets within the power system to manage constraints and support the integration of elastic energy resource (e.g., wind power and/or other intermittent renewable energy resources).

In particular implementations, the technique primarily uses two signals – the transactive incentive signal and the transactive feedback signal – representing the cost of power delivered to a given point in the system and the load at a given point in the system respectively. In particular embodiments, both signals are forward forecasts. The use of these representations reduces communications capacity requirements but relies on the development of algorithms for monetizing operational objectives. This was illustrated through a simple electric vehicle charging example and an extended example for wind power integration.

3 Exemplary Embodiments of the Disclosed Transactive Control Signals

3.1 Introduction

The transactive control and coordination system (TCS) of the disclosed technology can be implemented primarily using two classes of transactive signals: transactive incentive signals (TIS) and transactive feedback signals (TFS). These signals are exchanged between distributed system sites. The purpose of these signals is to coordinate supply and load in the near future, from a few minutes to several days out.

Some might compare the *TCS* with locational marginal pricing (*LMP*), in which energy prices are differentiated by time and by circuit location to address the economics of resource availability and to help mitigate transmission system congestion. A *TCS* shares certain goals with *LMP*. Like an *LMP* price signal, a *TIS* is a price-like signal that may represent the value of energy resources while taking into account the location, the time, transmission congestion, and transmission losses. Unlike an *LMP* signal, however, a *transactive signal* has been generalized to represent other additional impacts that can be monetized. Furthermore, a *TCS* facilitates fully distributed, not centralized, formulations of *transactive signals*. Because the calculations may be fully distributed, a *TCS* system is scalable throughout transmission systems, distribution systems, customer premises, and/or device levels.

An *LMP* represents the cost of the marginal energy resource and is therefore useful for coordinating the dispatch of energy resources. An implication is that dispatch decisions for supply-side or demand-side resources are based solely on comparison against the current marginal resource. By contrast, embodiments of the *TIS* are preferably formulated to represent energy *cost* as a function of time and location so that it may coordinate multiple supply-side and demand-side resources, not just the marginal ones. (This distinction is increasingly of interest as must-run renewable resources become a significant fraction of system resources. Economic dispatch and marginal energy price are currently based largely on fuel expenses. Renewable resources, which consume no fuel, displace fueled resources. Therefore, the marginal price, which is determined by the marginal fueled resource, incurs downward pressure. If the resulting marginal price is used to calculate revenues, then revenues also experience downward pressure, even though the must-run renewable resources may have generated relatively expensive energy.) The economic usefulness of many resources is determined during planning stages, not as they operate. Once the resource has been built, it should be called upon anytime it is useful, not only when it competes well with the current marginal resource.

A *TCS* and its *transactive signal*s, in principle, may thereby unify some decision processes that are conventionally addressed separately or sequentially—the using the dispatch of must-run resources and economic dispatch, for example, or the testing of economic power flow against permissible constrained power flow.

While quantity of energy is most certainly used during the calculations of *LMP* signals, there is seldom a need for those signals to be communicated outside the location of the central solver. In embodiments of the disclosed technology, however, the *TFS*, which represents a quantity of power, accompanies the price-like *TIS*. For example, distributed formulations can be used with signals that represent both the paired price and the quantity of power for time intervals. In particular, *transactive signals* can enable the coordination of the *TCS*, where each *transactive node* has a responsibility to perform its share of what is presently a very centralized calculation. The standardization of a *TCS* and its *transactive signals* can permit new implementers to join a *TCS*.

Now that some general characteristics of a *TCS* have been introduced, largely through a comparison between *TCS* and LMP systems (see, e.g., Table **1**), further details and qualities of the *TCS* will be introduced. For example, the sections below describe the component parts of a *TCS*, including its *transactive signals*, and how each of the two subclasses of *transactive signal* are influenced and formulated.

Table 1: Comparison Summary between *LMP* and *TCS*

LMP	*TCS*
Calculation is performed centrally	Calculation may be distributed
Signal represents unit price of marginal resource	Signals preferably represent inclusive unit cost of energy and quantity of energy
Somewhat scalable to disaggregated regions of generation, transmission, maybe into distribution	Very scalable, in principle, throughout generation, transmission, distribution, customer, and end-use devices
Usually relevant only to perspective of one single system operator	May represent perspectives of any and many system component owners
Contractually engages large blocks of firm resources	May engage many small, flexible resources and large blocks of firm resources alike through the normal course of energy pricing or through alternative and diverse incentive mechanisms
May include forecasted future intervals	Includes forecasted future intervals

3.2 An Example Transactive Coordination and Control System

An exemplary embodiment of the *TCS* may be understood by its components and their behaviors. In particular implementations, its principal components comprise one or more of the following:

- *transactive node*—system sites that are active participants in *a TCS*. A *transactive node* hosts a *transactive node object model* and exchanges *transactive signals* with its *transactive neighbors.*
- *transactive signal*—comprises one or more subclasses of signals that are exchanged by *transactive nodes.* For instance, in particular implementations, the transactive signal comprises two subclasses that include the *TIS* and *TFS*.
- *transactive node object model*—the state model of the actions and responsibilities that are managed by a *transactive node*
- *toolkit functions*—one or more functions that may be called upon by the *transactive node object model* to customize it for the unique set of inelastic and elastic supply and demand-side resources that are managed at a respective *transactive node*. The functions can belong, for example, to a plurality of subclasses. The subclasses can include, for instance, t*oolkit resource and incentive functions* and t*oolkit load functions*.

3.3 Example Transactive Node

In embodiments of the disclosed technology, *transactive nodes* are points in the topology of a *TCS*. In particular embodiments, *transactive nodes* periodically exchange *transactive signal*s with their neighbors (e.g., their nearest neighbors) with which they can exchange electrical energy. For instance, transactive signals are exchanged between *neighboring transactive node*s that share an electrical conductor. (This is true in the sense that two *transactive nodes* that exchange power also communicate. The actual pathway and communication media between *transactive node*s can vary from implementation to implementation.) The resulting interconnection topology can, in some embodiments, be ***hierarchical*. Transactive nodes can be established at any hierarchical point in the topology** (e.g., at any point of the utility-side topology, such as a sub-station, feeder, transformer, or the loke) or at any point of the load-side topology, a feeder, transformer, household control unit, electric vehicle charger, or any control unit at the household or other load control unit).

3.4 Example Transactive Signals

*Transactive signal*s can be represented as a series of data. For instance, in particular implementations, the transactive signals are a series of triplets. Each triplet is comprised of a time interval, a value, and a confidence level that qualifies the value. In other implementations, the transactive signals comprise a series of value pairs, where each value pair comprises any combination of a time interval, a value, or a confidence level. In still other implementations, the transactive signals comprise one or more of a time interval, a value, and/or a confidence level. In particular implementations, there are two subclasses of *transactive signals*:

- the *TIS*—a representation of preferably the delivered unit cost of the energy that is stated in the corresponding *TFS*. There is a *TIS* representation at each *transactive node* and for each time interval.
- the *TFS*—the power flowing between two *transactive nodes* during a given time interval. The unit cost of the energy that is being exchanged is the corresponding *TIS* of the given time interval and for the given *transactive node* that supplies the energy. There is a *TFS* representation for each *transactive neighbor* at each *transactive node* and for each time interval.

The examples herein were simplified to address real power and real energy. However, the reader skilled in the art of electrical power will understand that the examples could be extended to refer to real energy (meaning the product of real power and elapsed time), reactive energy (meaning the product of reactive power and elapsed time), or both real and reactive energy components. That is, a TIS may separately or jointly monetize real energy, reactive energy, or both real and reactive energies, and a TFS may represent real, reactive, or both the real and reactive power components of the power flowing between two transactive nodes.

3.4.1 Predictive Signal Intervals

In particular embodiments, the *transactive signals* are forecasts. The forecasts refer to an imminent time interval (e.g., the time interval that will start next) and a number of additional future intervals thereafter. The future intervals are defined by their starting times and durations. Once stated, an interval remains fixed in time, and a future interval moves closer with the passing of time. The intervals in a *transactive signal* are successive in one particular embodiment of the disclosed technology (e.g., they do not overlap).

A subsequent *transactive signal* updates the values and confidence levels for many or all of the previous *transactive signal's* time intervals. New intervals may also be created to push the forecast even farther into the future.

In one particular embodiment of the disclosed technology, termed "the demonstration", **56** successive intervals ranging in duration from **5** minutes to **1** day were elected. Refer, for instance, to Table **2**. It should be understood, however, that any number of intervals of any duration can be used to implement embodiments of the disclosed technology. In Table **2**, the term "IST_n" refers to the time at which the n^{th} interval begins—the *interval start time.* The durations of the thirteenth, thirty-third, fifty-first, and fifty-fifth interval may change from one *transactive signal* to the next; this was done in the illustrated embodiment to make sure that the intervals remain aligned with major **15**-minute, **1**-hour, **6**-hour, and **1**-day transitions.

The shortest interval could be any duration. For instance, the duration might be limited by the sum of the system's calculation and communication latencies. If the system were to use relatively short intervals (e.g., five minutes or less), it could respond to many dynamic issues, even area control errors, which are typically managed on **4**-second intervals.

In one embodiment, intervals were defined with increasingly longer durations into the future because more distant future values may only be meaningfully and accurately forecasted in a statistical, averaged sense. For example, if one knows the accurate status of a thermostat and the building temperature that the thermostat manages, one may accurately predict quite precisely when this system will begin or end its current heating or cooling cycle. For tomorrow, however, one cannot predict precisely when each cycle will begin and end, but one can quite accurately predict the fraction of time that the system will be actively cooling or heating. (In other embodiments, longer intervals (such as over **1** hour) are avoided. It has been observed, for example, that intervals longer than **1** hour tend to destroy important boundaries that have been defined at the boundaries between hours. For example, some utility billing practices presently distinguish "heavy load hours" that occur from **6**:**00** a.m. to **10**:**00** p.m. Pacific.)

The **56** intervals used in the example embodiment discussed herein extend more than **3** days into the future, but could extend to any desired time period. The total number of intervals and durations of the longest intervals in the example embodiment were influenced by the desire to allow the system to be unattended for at least three days—the duration of a long holiday weekend.

Table 2: Example Intervals

Duration	No. Intervals	Interval Start Times
5 minutes	12	IST_0, IST_0 + 0:05, ..., IST_{10} + 0:05
15 minutes	20	Round(IST_{11} + 0:15)*, IST_{12} + 0:15, ..., IST_{30} + 0:15
1 hour	18	Round(IST_{31} + 1:00)*, IST_{32} + 1:00, ..., IST_{48} + 1:00
6 hours	4	Round(IST_{49} + 6:00)*, IST_{50} + 6:00, ..., IST_{52} + 6:00
1 day	2	Round(IST_{53} + 1:00:00)*, IST_{54} + 1:00:00, IST_{55} + 1:00:00
>3 days	56 intervals	57 interval start times (IST)

*The function "Round" indicates rounding down to the next 15-minute, 1-hour, 6-hour, or 1-day interval start time. Times are indicated as dd:hh:mm (days, hours, and minutes).

In Table 2, the 57th IST was used to define the end of the 56th interval, which is the final interval in a *transactive signal* of the example embodiment.

Published future intervals remain valid and may be used, in principle, until they are overcome by time. This means that a *transactive signal's* Friday forecast for a Monday morning interval can be used even if the system fails to calculate any new *transactive signals* through the weekend. In this capability, the system is resilient to temporary failures of individual system components. If, however, a part of the system fails, the signals that had been predicted much earlier become increasingly dated and inaccurate. The system also loses its ability to recognize and respond to change while new signals are absent. Also, because later intervals have longer duration, signal dynamics diminish as the system relies on progressively longer prior predictions. In one embodiment, the confidence attribute is degraded (e.g., indicates diminished confidence) over time as signals become stale, unupdated.

Although any suitable time standard can be used, embodiments of the disclosed technology use the Coordinated Universal Time (UTC) standard (ISO/IEC 2004). The UTC can be used, for example, to enforce a consistent and standardized representation of time across time zones. UTC times are unchanged across time zones and across transitions into and out of daylight savings periods. In certain embodiments, and in order to avoid problems with aligning time zones ad contractual obligations that may exist, the use of intervals longer than one hour is avoided.

3.4.2 Confidence Attribute of a *Transactive Signal*

In some embodiments, transactive signals also include a confidence attribute that is specified to qualify the values in the *transactive signals*. In particular implementations, the confidence attribute estimates the relative positive root-mean-square (RMS) accuracy of each value that is published in a *transactive signal*. In many cases, this interpretation is quite naturally incorporated. For example, forecasts for renewable energy resources are already qualified in a way comparable to an RMS error.

Some events or conditions are not as naturally represented using the metric relative RMS error. For example, one might have diminished confidence if a signal has been delayed or if some component information to be used in a calculation has become stale. Other examples might include startup conditions while only limited information has been received, suspect status of computational equipment that hosts a calculation, or calculated values that are simply outside a normally accepted range for unknown reasons. Nevertheless, these conditions can be functionally represented by relative RMS error.

The recipient of a value that is accompanied by a high relative RMS error may use such information in many ways. The local practices and policies may differ at each *transactive node*. The possible responses include, for example, the publication of error or warning flags, performing alternative calculations that are more conservative, resorting to safe default values, using statistical algorithms that optimize outcomes or minimize risk, or no action at all.

3.4.3 Transactive Incentive Signal

In particular embodiments, a *transactive node* has one *TIS* for any given time interval and any given calculation result. No differentiation of *TIS* value is allowed across a *transactive node*. If for any reason electrical energy should be valued differently across a *transactive node*, the *transactive node* should be divided into more than one node at the feature that causes different valuation.

In one particular implementation, the *TIS* is calculated by summing the incurred costs and dividing the sum by the energy to which the costs refer. The total energy may be thought of as either entire load (including exported energy), or as the entire supply (including imported energy), at the *transactive node*. The *transactive node* can assume that total supply is equal to total load. It has been found that it is more natural to work from the supply side during the formulation of *TIS*. It is the costs of the various mixes of supply resources that directly affect the *TIS*.

The input parameters of the TIS formula in Table 3 create a useful interoperability boundary. The parameters represent various costs ("C") and power ("P"), where the subscripts refer to terms for energy ("E"), generation ("G"), capacity ("C"), infrastructure ("I"), or other ("O"). Further, subscript n is the interval number and Δt_n is that interval's duration. Members of a *TCS* may be invited to generate their own functional algorithms that in turn influence the *TIS* by simply designing algorithms that assign values to these various parameters. The parameters are distinguished by their units. Implementers may select and use the parameters that most naturally represent the forecasted cost impacts. It should be understood that these parameters are not limiting or even required for a particular component. In certain embodiments of the disclosed technology, the functions that generate these parameters are called *toolkit resource and incentive functions*. Resource functions model energy supply resources. Incentive functions affect the *TIS*, but they do not represent any energy resource. Example resource and incentive functions are described in more detail below, including Appendices B and C.

Table 3. Example formula by which the *TIS* is to be updated

$$TIS_n = \frac{\sum_{a=1}^{A} C_{E,a,n} \cdot \hat{P}_{G,a,n} \cdot \Delta t_n + \sum_{b=1}^{B} C_{C,b,n} \cdot \hat{P}_{C,b,n} + \sum_{d=1}^{D} C_{O,d,n}}{\sum_{a=1}^{A} \hat{P}_{G,a,n} \cdot \Delta t_n} + \sum_{c=1}^{C} C_{os,c},$$

Or

$$TIS = \left(\frac{\textit{energy cost} + \textit{capacity cost} + \textit{other costs}}{\textit{energy resources}} \right) + \textit{offset costs}$$

In other embodiments, infrastructure costs are among the numerator terms. However, in such embodiments, an undesirable inverse relationship between TIS and total power demand may result. In Table **3**, infrastructure costs can be included among the "offset costs".

3.4.4 Transactive Feedback Signal

The *TFS* is calculated readily for a radial distribution circuit branch. The *transactive node* on a radial distribution branch simply sums its predicted inelastic and elastic loads. The upstream *transactive node* is the only resource available to supply the load at this system location, so the *TFS* is identical to the predicted load for the branch.

The *TFS* is not as easily predicted between *transactive node*s that are not on a radial distribution branch and have more than one *transactive neighbor*. Their network system connections may be *meshed*. Desirably, power flow is allocated among multiple *TFS* in a way that would be fully consistent with a proper power flow calculation.

In a fully deployed *TCS*, economic dispatch decisions would be made at each transactive *node* to balance load. To the degree that energy can be imported from the *transactive node's neighbors*, the *neighbors*' energy competes with local resources. Any mismatch is desirably allocated among the *TFS*s.

In certain embodiments, each member of a pair of *transactive neighbors* estimates a *TFS* for the interface that they share. (The general case of meshed networks and bidirectional power flow desirably uses each *transactive neighbor* to publish and receive paired cost (TIS) and quantity (TFS) signals.) The convergence of the two estimates is a metric that can be used to determine whether the two neighbors have concluded their negotiated solution or not.

3.5 Transactive Node Object Model

In certain embodiments, the formal model of the *transactive node* class and its behavior has been specified by the *transactive node object model.*

3.5.1 Algorithmic Framework

An example model of the algorithmic responsibilities of a *transactive node* is introduced below in Appendix B. The details of this model can be used to implement exemplary transactive nodes (e.g., using Standard ISO/IEC **18012** (ISO/IEC **2004**) or using a unified object-oriented modeling language such as UML-**2** (OMG **2013**)). The algorithmic framework has proven to be applicable across many different types of *transactive node*s.

Figures **12A** and **12B** are a skeleton diagram **1200** of the algorithmic framework at a *transactive node*. The diagram addresses two main objectives: First, it provides that the *TIS* may be calculated. Second, it provides that the *TFS* may be calculated.

A particular implementation of the function "**3**. Formulate *TIS*" is disclosed in Appendix B. This function receives information about intervals, costs of various resources and incentives, and the sum of imported and generated energy to which the cost information is relevant.

The model states that both the input information and resulting *TIS* values are stored in a data buffer. These buffer contents may be mined for data by those who have permission to do so. But the greater importance of the buffered data is that such stored information makes the system resilient to imperfect communications: the input values from a prior series of forecast intervals remain this *transactive node*'s best prediction of the input interval values until updated information can be received. This is especially useful when the information is delayed or when a communication link becomes temporarily severed.

The impacts of energy supply and incentives (or disincentives) at a given *transactive node* are received through *toolkit resource and incentive functions*, a modular library of functions that model the costs and energy supplied by energy resources and other cost incentives or disincentives at a given *transactive node*. An example implementation of the function "**8**. Calculate Applicable Toolkit Resources and Incentives" (near the top center of Figures **12A** and **12B**) is disclosed in Appendix B. In certain embodiments, these *toolkit functions* are not themselves inside the algorithmic framework, but they inject their influences into the updating of the *TIS* via a standardized set of parameters.

A particular implementation of the function "**4**. Formulate *TFS*" (at the bottom right of Figures **12A** and **12B**) is disclosed in Appendix B. The objective of this algorithmic framework function is to forecast the flow of energy between it and its *transactive neighbors*. It therefore receives information about the set of future intervals. It also receives information about forecasted supply and load so that the balance may be allocated to the *TFS* between this *transactive node* and its *transactive neighbors*.

In certain embodiments, the load forecast has two threads. The first forecasts the inelastic load. This is the base case that is unaffected by the *TIS*. The second thread is the elastic load—the change in load that may be attributed to the *TIS* and events that are generated in light of the *TIS*. The separation of these threads is practical and it helps measure and verify system responses. The sum of the inelastic and elastic load forecast components accurately forecasts the actual load.

Table 4. Formula for total load used for TFS

Total load = Inelastic load + Change in elastic load

The model of a single asset system may forecast both inelastic and elastic load components. For example, the thermostatic building asset model forecasts both its normal building load and the changes in load caused by temperature setback events. In certain embodiments of the disclosed technology, a single feeder model forecasted bulk inelastic load that in effect included many inelastic components of responsive assets. Provided that the components are properly summed for the given *transactive node* and not double-counted, it will not matter that the thermostat model did not model its own inelastic load component.

More information about the *toolkit resource and incentive* and *toolkit load functions* are discussed below as well as in Appendices B and C.

3.5.2 Signal Timing

In certain embodiments, the *transactive node object model* includes functionality and attributes that control the times at which *transactive signal*s are transmitted to *transactive neighbors*. An exemplary timing model is discussed in this section, but is not to be construed as limiting, as any number of intervals having other durations can be used. The example timing model was designed to allow propagation of information about disturbances (e.g., of the electric transmission grid) across the *TCS* system while reducing unfruitful chatter and calculations. As noted, the example timing model is not necessarily one that should be standardized or used in implementations of the systems.

A *transactive node* should normally not publish *transactive signal*s for which any interval starting time has already passed. This expectation creates a useful framework for the calibration of system clocks. The error between clocks at different system locations should desirably be small compared to the shortest intervals—**5** minutes for the example timing model. Tight tolerances are, in principle, achievable for *transactive node*s that are internet connected.

In the example timing model, each *transactive node*, at the beginning of a **5**-minute interval, publishes *transactive signals* that address the interval that begins **5** minutes from now and into the future.

Various timers were implemented to avoid unnecessary chatter. One timer begins when a *transactive signal* is received. Another timer begins after a *transactive signal* is transmitted. No *transactive signal* of the same type may be transmitted again until after these timers expire. Figure **13** is a block diagram **1300** illustrating the example timing model.

In one embodiment, the time model is event-based. For example, the timing model can be adapted to become more responsive to status or condition events and less reliant upon clock-based events (e.g., hold-down timers, interval timers). New signals and additional calculations can be generated only after significant changes occur to schedules and forecasts, either locally or at remote system locations. As long as forecasts remain accurate, the system should be unperturbed.

Further, sets of prediction intervals that are nested rather than sequential can be used. That is, an understanding that the next **5** minutes are a subset of an hour-long interval that is a subset of the day that is a subset of a month, and so on, can be adopted.

Still further, in some instances, a *relaxation criterion* against which forecast changes may be compared can be used. The criterion can state a weighting of errors for each interval. For example, if the sum of the errors exceeds the overall threshold for a *transactive signal*, then the signal is updated and republished; otherwise, no signals should be transmitted because the changes are deemed to be insignificant. This criterion can be used in an event-based model wherein imminent and future intervals are rapidly iterated (e.g., on an asynchronous basis) until they resolve according to this criterion.

3.6 Transactive Data Collection Layer

In some embodiments, a transactive data collection system layer is also defined and used in implementations of the transactive nodes. For example, this system layer automatically retrieves toolkit function outputs from *resource, incentive*, and *toolkit load functions*; gathers resulting *TIS* and *TFS* signals that are generated at each node from its *toolkit function* inputs; and records various system management events and statuses. Because the system is distributed both in time and space, it is desirable to keep track of data provenance, including locations of nodes from which the data originates, times at which signals are generated, and time intervals to which predictive signals refer.

One advantage of a *TCS* is that the *transactive signals*, while revealing an aggregated cost and quantity of energy, do not necessarily reveal any sensitive or private data. The model used to store and collect information about local resources and loads at a *transactive node* can be useful, but such information would normally be shared only with the owner of a set of *transactive nodes*, who is entitled to receive such privileged information. Desirably, little or no sensitive information is shared by *neighboring transactive nodes*.

"*Non-transactive*" data can also be defined and collected. *Non-transactive* data is factual data that is collected from system meters and which can be used during analysis to assess the success with which the predictive *TCS* has influenced system loads and its consumption of various energy resources. *Non-transactive* data can also include weather data at each distributed site.

3.7 Influences on the *TIS*

This section addresses the formulation and interpretation of the *TIS*.

3.7.1 The *TIS* is an Aggregate of Multiple Resource and Incentive Costs

In some embodiments, while each *TIS* states a value for each future interval, each said value may be composed of a plurality of various resource and incentive cost components. This concept is demonstrated by diagram **1400** in Figure **14**, which shows multiple stacked component costs, the sum of which is the published *TIS* value. The biggest cost component, in this example, is the unit cost of the energy that is received by this *transactive node* from its *transactive neighbors* (the transactive component). The remaining components are ranked as the cost of infrastructure and the unit costs of wind, hydroelectric, and fossil-fueled resources.

Observe that influences are inherited from neighboring *transactive nodes* that supply this *transactive node*. For example, if 8% of a *TIS* value is from the costs of fossil energy resources, and if this *transactive node* is supplied another 10% of its resources by a neighbor for which 10% of this neighbor's *TIS* value is from fossil resources, then the total impact of fossil energy on the *TIS* at this *transactive node* would be 8% + 10% × 10% = 9%.

Therefore, one can look to propagated resource mixes one, two, or even more neighbors distant to accurately assess the resource supply mix at this *transactive node*.

3.7.2 *TIS* Calibration Measurements Identified

As discussed, in certain embodiments, delivered cost of energy is used as the metric for *TIS* magnitude. This metric is useful because (**1**) it provides a straight path to using the signal for revenue, if other implementations choose to do so, and (**2**) comparable calibration standards exist at some locations within a *TCS* for this metric.

In a distributed system, checks and balances are desirable to make sure that the *TIS*, which is collaboratively formulated, is meaningful and fair. The first step toward accomplishing this was to establish a common semantic understanding of the *TIS* as, for one embodiment, the delivered cost of energy at a location. The second step is the comparison of the *TIS* and its components against comparable calibration standards. For example, existing and historical contracts define the average unit cost of energy among many suppliers and recipients of electrical energy. Distribution utilities can accurately state how much they paid for a unit of energy during the past year. Therefore, the *TIS* and any other valid representation of the delivered cost of energy at a system location should be comparable over long periods of time.

3.7.3 Resource Toolkit Functions

Adequate energy resources are desirably received into or dispatched at a *transactive node* to balance system load. The mix of dispatched energy resources can be determined in a distributed manner (though it is also possible to use a central determination for smaller scale implementations).

In certain embodiments, *resource toolkit functions* from a library of functions are the functions that calculate the quantity of energy and its cost impacts toward the formulation of the *TIS* at a *transactive node*. The *resource toolkit functions* can reside at any of the transactive nodes (e.g., *transmission zone nodes*, which each represent large regions of a region's generation and transmission systems). One or more of the following functions can be used to represent groups of (or individual) energy resources:

- *Non-transactive energy* function—represents energy imported into the system from entities that are not *transactive node*s.

- *Transactive energy* function—represents energy imported from a *neighboring transactive node*.
- Wind energy function—represents energy from wind farms in this *transactive node*.
- Hydropower generation energy function—represents energy from hydropower at this *transactive node*.
- Fossil generation energy function—represents energy from fossil (more generally, “thermal”) resources at this *transactive node*.
- Solar energy resource function—represents energy from solar resources at this *transactive node*.

3.7.4 Incentive Toolkit Functions

Incentive functions are similar to *resource functions*, but they are not tied to energy supply. One or more of the following exemplary incentive functions can be used in implementations of the disclosed technology:

- *Transmission congestion* management function—if the power flowing through electricity transmission lines between two transactive nodes ever approaches the capacity limit on the transmission lines, this function adds cost disincentives to the downstream *transactive node* to reduce load on the line.
- Cost of general infrastructure function—a cost that is amortized over time to represent the cost impacts of built infrastructure that has not otherwise been captured in the system. The offset from this function calibrates the *TIS* over time, pulling it gradually toward a reasonable *TIS* at each *transactive node*.
- Demand charges function—this is an *incentive toolkit function* that can be applied at *utility-site* transactive *node*s. Wholesale electriticy suppliers charge their utility customers according to quite complex cost structures. This function attempts to represent the cost impacts of demand charges and, to a lesser degree, time-of-use charges. Functions have been drafted to represent the cost structures of, for example, regional power administrations.

3.8 Influences on the *TFS*

A *TFS* represents the power flowing between a *transactive node* and its *transactive neighbor* during the imminent and future intervals. The majority of the power flow is usually *inelastic*: it is unaffected by the predicted unit cost of the energy—the *TIS*. If the *transactive node* hosts responsive asset systems, these systems might observe the *TIS* and change their forecast of how much energy they will consume during a future interval—they are *elastic*. The *transactive node state model* keeps track of the changes in load that are anticipated from these elastic asset systems.

Responsive asset systems that curtail load reduce load at a *transactive node* and therefore tend to reduce the energy that is generated at or imported into the *transactive node*. Demand-side generators have the same impact when they generate energy and displace load at the *transactive node*.

Even more useful are responsive asset systems that can *increase* their energy consumption (or equivalently, *reduce* their demand-side generation). These asset systems thereby increase system load at their *transactive node*s and increase the energy that is either generated at or imported into the *transactive node*. This response is increasingly useful in power grids that experience excessive generation, as now occurs in regions that have high wind-power penetration.

3.8.1 *TFS* Calibration Measurements Identified

A straightforward comparison standard exists for *TFS* values at many system locations. Because the *TFS* represents forecasted power flow, the accuracy of the forecasted power-flow values in a *TFS* may be compared against actual metered power at that point in the power grid. For example, the electricity supplied to a distribution by its electricity supplier is accurately metered.

3.8.2 Inelastic Load Prediction Functions

Inelastic load functions forecast baseline load that is unaffected by the *TIS*. Inelastic load functions can be defined for each residential, commercial, and industrial load type. The load from these models can be scaled by the numbers of each customer type. Alternatively, a parametric model can be used that can be trained by historical data. The model appears to perform similarly for all of the different load types. The forecast model creates a correlation to forecasted weather information—including at least ambient temperature. If available, the model can also incorporate recent measurement data to improve the forecast.

3.8.3 Elastic Load Functions

Elastic *toolkit load functions* in conjunction with *asset models* model how responsive asset systems are influenced by the *TIS.* In certain embodiments, these functions have two principal responsibilities: First, the *toolkit load function* predicts when events may occur and how long they will last. Second, an *asset model* forecasts the change in load that will occur during an event for the given asset system.

Elastic toolkit load functions can be categorized as follows based on the nature of their forecasted events:

- Event-driven—several events may be called each month. The principal challenge is to allocate a limited number of allowed yearly, monthly, and daily events (e.g., curtailment events) based on the forecasted *TIS*. Additional restrictions may apply to the minimum and maximum durations of the events for a given asset system.
- Daily events (sometimes referred to herein as “time-of-use” events)—events are expected to occur almost daily. The events might be specified differently for weekdays, weekend days, and holidays. The principal challenge is to place an event at the best time of day based on the *TIS*. Additional restrictions may apply to the minimum and maximum durations of the events for each day type.
- Continuous—in some emobdiments, dynamic responses are being made every interval. The challenge is not so much to specify events as to state a functional relationship between each *TIS* value and a system response.

An *asset model* then models the change in load during the above event types. It has been found that many possible pairings exist between event types and *asset model* types. For example, a water heater *asset model* may be used with either event-driven or daily event types. In principle, water heaters could be manufactured to have continuous responses.

By way of example, one or more of the following exemplary *asset models* can be applied in an implementation of the disclosed technology:

- water heater population—for instance, the population of residential **40**-gallon water heaters controlled by in-line switches (e.g., demand-response units). (Models for other sizes of water heater can also be used.) After the timing of events has been predicted, the challenge is to predict the power and energy that will be curtailed by the systems response.

- thermostatic space conditioning with temperature setback—in one implementation, a first-order thermal model of a building is simulated. The model is scaled by numbers of building types and their thermal properties, parameters which are desirably configured by the implementer of this elastic *toolkit load function*. Dynamic inputs include ambient temperature, solar insolation, and modeled target interior temperatures that represent occupancy temperature settings. During events, the modeled target temperature is raised or lowered, depending on whether it is cooling or heating season. An advantage of using this thermal model is that it predicts thermal rebound if buildings that have had their thermostat load set back return to normal operation.

- thermostatic space conditioning with cycling of the heating, ventilating and air-conditioning (HVAC) unit—uses the same first-order thermal model and simulation as for temperature setback, but events cause a reduction in modeled power of the space conditioning equipment to represent the cycling of HVAC equipment.

- stationary battery storage systems—the *TIS* is an input to a simulation model that attempts to maximize the cost of energy discharged into the grid and minimize the cost of energy used to charge the batteries. The exchange of energy is scaled by and limited by the modeled useable energy capacity of the batteries and by the capacity of the bidirectional power converter that charges and discharges power into and out from the batteries. The responsiveness of the system may also be modified depending on how frequently the system's owner will permit it to become alternately charged and discharged.

- controlled distribution voltage systems—estimate the change in load that will accompany a change in distribution voltage during an event. In one simplified implementation, the asset model uses a static factor to represent the change in load as a function of change in a feeder's voltage.

- distributed generators—models a change in generation during events. In most cases, the generator becomes activated during events, and the generator supplies its nameplate rated power or another prescribed power level during the event.

- in-home display and portal notifications—in one implementation, event periods are presumed to be indicated to in-home displays or portals as a small number of discrete states (e.g., a high-price event). The change in load is, of course, dependent upon the election of a population of energy customers to voluntarily turn their devices on or off. A typical change in power is forecasted by time of day that may be scaled by the numbers of in-home displays or portals in the population.

- a suite of smart appliances, including washer, dryer, and dishwasher—These appliances are similar to in-home displays in that they notify customers of events during which the smart appliance owners may may elect to defer electrical load. In another exemplary implementation, these appliances have additional features by which customers may better automate decisions to delay the appliance loads, and some energy reduction is also achieved automatically when the appliances are in their conservation mode. The change in load is modeled simply as a fraction of typical appliance load by time of day and by appliance type.

Table **5** summarizes the potential pairings of the listed exemplary *asset models* with appropriate event types. Examples for some of these pairings are described in the appendices below. Implementations for other pairings can be developed by those skilled in the art in view of this disclosure.

Table 5. Pairing of Response Characteristics with Asset Models

Asset models	Event-Driven	Daily Events	Real-Time
Water heaters	Y	Y	Y
Thermostat setback	Y	Y	Y
HVAC cycling	Y	Y	Y
Battery storage	Y	Y	Y
Distribution voltage control	Y	Y	Y
Distributed generators	Y	Y	Y

In-home displays/portals	Y	Y	Y
Other smart appliances	Y	Y	Y

3.9 Additional Observations

In a fully deployed *TCS*, regional transmission and generation owners formulate *TIS* signals by stating the temporal and locational value of resources at many transmission and generation sites in the region, and the *TFS*, a feedback signal, influences their resource dispatch decisions at these distributed locations.

Further, as household devices become more intelligent, there will eventually exist vast populations of flexible, responsive assets that would be active in a *TCS*. These assets will be available to modify their consumption at each update interval. A *TCS* invites the demand side to participate in the system objectives on equal footing with supply.

3.10 Interoperability

Implementations of the disclosed technology can be standardized, if desired. Standardization efforts may be at a variety of different levels. For instance, the TCS can be defined at the organization and informational level. In this regard, Figure **15** is a diagram **1500** showing an example skeleton model of a standard *transactive node* and the signals that it communicates with other *transactive nodes* and with modules and systems, some of which can be outside the boundary of a standardized system. Typically, neighboring *transactive node*s will have to agree between themselves concerning an Interoperability Framework, including the remaining interoperability levels (“Technical/Syntactical” levels). Between unrelated sites, this negotiation is unique. However, if neighbor nodes share the same owner, a common technology may be applied to all the owner’s *transactive node*s. The *TCS* standard should desirably be agnostic of the technologies by which it may be implemented.

Certain implementers can choose to define additional implementation details beyond those in the standard. The implementations might, for example, further specify the syntactical levels of interoperability. These implementations should abide by and make reference to the main standard. However, the new implementations may themselves become standards, or they may be recognized as reference implementations of *TCS*.

Further, implementers may desire to keep their particular code (e.g., code for a toolkit function) confidential. Such a scenario is feasible so long as the resulting signals are conformant.

Embodiments of the disclosed technology can be integrated with academic distributed control approaches. For instance, the specification of *transactive signals* can be harmonized with signal characteristics specified in simulation studies. An outcome of such harmonization will be that the *transactive signal* that represents power flow will be a complex representation. (This use of *complex* here is mathematical. A *complex* number has real and imaginary components. The real component represents real power; the imaginary component represents the flow of reactive electrical power.)

Embodiments of the disclosed technology can be harmonized with LMP approaches. For instance, the practices of *LMP* and *TCS* can be harmonized, potentially allowing the TCS approach to compete with, supplement, or gain equal footing with *LMP* practices.

Embodiments of the disclosed technology can also be harmonized with other TCS approaches. For example, the price-like signal used in embodiments of the TCS approach may be modeled after cost, price, or competitive bids.

4 Overall Design for Embodiments of the Disclosed Transactive Control and Coordination System

In this section, additional details concerning the overall design for embodiments of a transactive control and coordination system according to the disclosed technology will be introduced. The discussion below also provides a supplemental discussion of the transactive control signals themselves. This discussion may, in some instances, be repetitive to the discussion above but is included herein for the sake of completeness.

4.1 Architecture of an Installed System Design

The architecture of an installed system is more diverse than for typical computer network designs. For instance, an installed system comprises generation, responsive assets, the electricity transmission and distribution systems, and digital communication and intelligence. The system therefore should consider:

- Physical, geographical location

- Electrical connectivity
- Information flow.

These components are interdependent, and a close correlation will typically exist and be maintained between them.

4.1.1 Physical, Geographical Architecture

The physical, geographical system architecture captures the physical locations of each piece of the installed system. Physical location can be influential to transactive control because local attributes (e.g., weather) affect the behaviors of equipment, end users, and responsive assets. One tenet of transactive control is that the value of supplied electrical energy is location-dependent. Physical, geographical architecture is easily captured on a conventional map.

4.1.2 Electrical Connectivity Architecture—the Nodal Hierarchy

The electrical connectivity system architecture captures the flow of electrical energy through the installed system. One tenet of transactive control is that the communication of value and operational opportunities (e.g., the transactive signals) in a transactive control and coordination system should logically follow the pathways of electrical energy flow. Existing and future power capacity constraints are highly path-dependent.

In certain embodiments, the electrical connectivity within an installed transactive control and coordination system forms a hierarchy of nodes. Here, the word *hierarchy* refers to a flow direction of electrical power and is not necessarily a static assignment. Electrical transmission systems are typically mesh (not radial) systems, meaning that parallel paths in the transmission system compete to supply load. The direction of electrical power in the transmission system may change. Some of this complexity will not be discussed in detail herein because embodiments of the disclosed technology can be adapted for such complexities using software tools that properly model meshed transmission power flow.

4.1.3 Information Flow in the Transactive Control and Coordination System

The information flow design captures the flow of data and information within an installed system. An information flow architecture also indicates where manual and automated decisions are made. The information flow architecture can include, for example,

- The communication channels used to transport transactive control signals
- The communication channels used to transport asset control signals
- The communication channels used for other data that supports local, regional, and client-run experiments
- Meter data channels through which meter data flows
- Locations within the information flow where functional calculations, like the estimation of future electrical load, take place
- Any other communication channels necessary to employ the installed transactive control and coordination system.

The information flow architecture can also capture details about the communication channels and signals, including communication media, protocols, bandwidth, formats, software tools, exemplary functional computations, and security attributes and practices.

4.2 A Generalized Transactive Control and Coordination System

This section introduces embodiments of hierarchical transactive control that can be used in an installed system. Prior to recent efforts to build a smarter grid, most all opportunities to manage and control electrical power have been managed quite centrally from the supply side—bulk electrical generation and transmission. The role of the power grid has been simply to satisfy electrical demand—the energy consumption patterns of all the end users. Embodiments of the smart grid according to the disclosed technology will engage end users and responsive assets throughout the grid, resulting in a cooperative, more distributed approach. Transactive control can facilitate this migration to a smarter grid.

4.3 Review of Transactive Control

Transactive control is a bidirectionally negotiated system behavior. Market-like principles facilitate the negotiation; however, the signals need not be used to account for any monetary or revenue exchanges. In theory, the “winning” behaviors are optimal in some sense, having competed successfully in a “market” against alternative actions that could have been taken.

One or more of the following are characteristics that can be exhibited in embodiments of a transactive control and coordination system according to the disclosed technology:

- Bidirectional communication—transactive control differs from the similar practice of real-time nodal pricing in that it uses dynamic feedback from its end uses.
- Incentives and feedback are communicated via one nodal signal—at a node, a single incentive time series is transported downstream, and a single feedback time series is transported upstream. Components of the incentive and feedback signals are additively combined into one incentive and one feedback time series. Using a single signal facilitates interoperability between multiple operational objectives and multiple responsive asset systems.
- Multiple operational objectives and responsive assets are simultaneously engaged—unlike present programmatic approaches that create unique engineered couplings between one operational objective and one or more responsive assets. Because operational objectives can be integrated into a single incentive time series, transactive control enables each responsive asset to respond to the integrated set of operational objectives. As a corollary, each operational objective may be acted upon by many responsive assets.
- The signal in a transactive control and coordination system can be dynamic on multiple time scales—transactive control signals are dynamic. In principle, the time intervals may be made infinitesimally small. A transactive control and coordination system could respond to a need for fast grid regulation, for example, if its time intervals were made short compared to the dynamic performance of fast regulation services. Regardless, the responsive assets may respond according to each asset's own dynamic capabilities and limitations. Not all parts of the system need to agree on and use the same interval if dissimilar interval signals can be added or interpolated to create valid comparisons between signals that have dissimilar intervals.
- Interoperability is facilitated—transactive control facilitates interoperability at the organizational and informational levels, and it allows technical layers of interoperability to become satisfied by any, or many, appropriate standards. This attribute helps make transactive control a worthy candidate for interoperable, regional smart grid communications.

- Responds **24/7**—transactive control can be always active. Small improvements and responses can be made throughout a day, not only during the worst several hours of the year.
- End-user friendly—by taking advantage of numerous short intervals and distributed digital intelligence, impacts on end users can be reduced, if not entirely eliminated. For instance, end users should have a final say concerning their comfort and should be provided options to temporarily opt out of responses.
- Facilitates distributed control—transactive control facilitates distributed intelligence and control. Centralized control is reduced or eliminated.
- Uses low bandwidth—the elimination of unique signals and the distribution of control should reduce communication bandwidth.

The transactive control technique of this disclosure can be compared to other approaches to transactive control, specifically the GridWise® Olympic Peninsula Project. Table **6** summarizes the major differences between the transactive control approach used during the Olympic Peninsula Project and embodiments of the disclosed transactive control approach.

Table 6: Comparison between the GridWise Olympic Peninsula Project and Embodiments of the Disclosed Technology

	GridWise Olympic Peninsula Project	Embodiments of the Disclosed Technology
Electricity customer incentives	Combinations of fixed and various dynamic price accounts. The project maintained a shadow market and customer accounts that were separate from utility billing.	Various approaches, as will be determined individually by participating utilities. Incentive practices should be sustainable.
Feedback signal	A bid was received from every responsive asset every five minutes ($/MWhr).	Each transactive node reports feedback that consists of a time series of expected energy consumption during each time interval into the future (kWhr/interval).

	GridWise Olympic Peninsula Project	Embodiments of the Disclosed Technology
Operational objectives addressed	One single transmission line constraint was addressed.	Multiple constraints, regional renewable energy availability, economic dispatch of resources, hydrogeneration, peak load mitigation, balancing resources, spot-market purchase mitigation, ...
Future time horizon	Not more than five minutes.	To be determined (probably from one to two days).
Approach for resolution of the "market"	Explicit clearing of the two-way "market" conducted every five minutes.	Uses iterative resolution of the "market" future intervals over time.
Shortest time intervals supported	Five minutes for real-time price customers.	To be determined (perhaps five minutes).
Architecture	Centralized. Information flow was managed from a central operations center and included the aggregator's communication servers.	Enforces a nodal hierarchy, including plans for standardization and extensibility of the hierarchy. Launched at multiple initial transactive node sites.

Exemplary components of embodiments of the transactive control and coordination system include one or more of:

- Transactive nodes—a physical point within an electrical connectivity map of the system. Electrical energy flows through a transactive node. A transactive node is not to be confused with locations within the information flow map that might also be called "nodes."
- Transactive signals—each node location receives an incentive signal from upstream and generates a corresponding feedback signal to be sent back upstream. These two signals—the transactive incentive signal and its feedback—together are the transactive signals.

- Responsive assets—the "prime movers" of the transactive control and coordination system that can modify consumption of electrical energy (e.g., in response to the current values of the transactive signals).
- Enabling assets—assets like communication infrastructure and metering that cannot by themselves modify energy consumption. Cost-benefit analysis typically cannot be properly assessed for an enabling asset alone because it represents only costs but no measureable smart grid benefits. The expenses of enabling assets are desirably allocated among and borne by truly responsive assets.

Responsive and enabling assets are more thoroughly discussed below.

4.4 Transactive Signals

This section describes example transactive signals and their use by the demonstration.

4.4.1 Introduction to Transactive Signals

In certain embodiments of the disclosed technology, there are two transactive signals at each transactive node:

- A transactive incentive signal (TIS) time series comprising the aggregated present and future values of the electricity supplied to and through each transactive node; and
- The transactive load feedback signal (TFS) comprising the sum of an estimate of the future quantity unresponsive and responsive electrical load to be consumed by the entire load downstream from the transactive node.

Each of the two signals is a time series, meaning that each is a vector of numbers, one for the present time interval and others for each future time interval (e.g., at least a day into the future). The time interval and horizon into the future can vary from embodiment to embodiment. In some embodiments, the time interval is five minutes. Shorter intervals than this would permit the demonstration system to provide additional ancillary services. Further, in some embodiments, shorter intervals are used for the near term and longer intervals into the signals' future. The signals' time horizon desirably extends at least to the future time when resource dispatch decisions are being made for the region.

The transactive signals at time t_0 can have the forms:

$$TIS = \{TIS(t_0),\ TIS(t_0+\Delta t),\ TIS(t_0+2\Delta t),\ TIS(t_0+3\Delta t),\ \ldots,\ TIS(t_f-\Delta t)\}$$
$$TFS = \{TFS(t_0),\ TFS(t_0+\Delta t),\ TFS(t_0+2\Delta t),\ TFS(t_0+3\Delta t),\ \ldots,\ TFS(t_f-\Delta t)\},$$

where *TIS* and *TFS* are the transactive incentive and feedback signals, respectively, Δt is the selected time interval, and t_f is the end of the prediction time horizon. The given time signal series can be updated next at time $t_0+\Delta t$.

The time-series elements of these two transactive signals are paired for each future time interval. This pairing between transactive incentive signal and transactive feedback signal is illustrated in block diagram **1600** of Figure **16**, which also portrays how an upward trend in the transactive incentive signal for any future time interval should result in a corresponding downward trend in load supplied through the transactive node for that time interval. If the transactive node supplies any responsive electrical load (e.g., responsive assets that are responsive to the transactive incentive signal), the responsive electrical load should respond to changes in the transactive incentive signal. Figure **16** indicates further that the granularity of the intervals for these signals could be relatively fine in the near term and courser into the distant future.

During the application of transactive signals, sensibility checks and default behaviors are desirably planned. For example, the nodes can be provided some independence to recognize and discount nonsensical signals that are believed to be erroneous. When no signals are received by transactive nodes, as may be the case when there has been a problem or equipment failure somewhere in the system, the nodes should again have the independence to revert to safe, bounded behaviors.

4.4.2 Transactive Incentive Signal

In particular embodiments of the disclosed technology, the transactive incentive time series is the main transactive signal. Each transactive node will typically have a unique blend of energy suppliers, upstream transmission pathways and distances, operational practices, local infrastructure, and/or downstream customers. Therefore, the values of the transactive incentive signal can be unique at a transactive node in the system.

In certain implementations, the basis for the transactive signal series at any node is a weighted sum of the transactive incentive signals received by that transactive node from immediately upstream transactive nodes that supply it electrical energy. The default approach, for example, can be to weigh the transactive signals according to the relative fraction of the node's power that is supplied from each upstream source as described below.

Each transactive node can also modify the transactive incentive signal that it relays downstream. At each transactive node, local conditions are analyzed and the incentive signal modified (or left unchanged) based on the local conditions. Modification of the incentive signal is for the purpose of influencing the behavior of responsive assets downstream from the node. The basic action at any node can be simply represented as:

$$\boxed{TIS_{output}(t) = \text{Weighted average}(TIS_{input}(t)) + \text{New incentives(t)}}$$

$$\boxed{TIS_{output}(t) = \text{Weighted average}(TIS_{input}(t)) + \text{New incentives(t).}}$$

Examples of how and why a transactive node will modify its transactive incentive signal include:

- The expense of energy supplied at the node—those transactive nodes that host generation have the opportunity and responsibility to insert the initial incentive signal values for that resource. For example, the incentive signal may reflect fuel expenses, infrastructure expenses, and/or all other expenses that are incurred to operate the resource. Ideally, the sum of incentives inserted for a generator over a year or longer should approach the sum of its true operational expenses.

- Infrastructure constraints or congestion avoidance imposed by the node—if the node itself becomes electrically constrained, it should modify the transactive incentive signal to incentivize downstream behavior that will alleviate the constraint. For example, the modification might be set equivalent to the incremental expense that would be incurred from the consequent shortening of a piece of equipment's lifetime, plus the likelihood that expenses will be incurred from outages after exceeding equipment ratings.

- Amortization and other expenses of installed equipment—even idle equipment can be argued to incur expenses. One should insert an expense for maintaining necessary infrastructure of the node. This incentive component, for example, is part of a natural disincentive for consuming energy far from where it is generated and thus using transmission infrastructure.

- Energy losses—modifications of the transactive incentive signal may account for line, transformation, and equipment energy losses.
- Operational objectives that occur at business entity boundaries—especially at business entity boundaries, the system shall encounter new operational objectives and values that should be respected. For example, certain utilities manage spot market purchases that are not influential in the regional hierarchy but become important at the boundaries of that utility.

The formulation of the transactive incentive signal can, but need not directly, incorporate actual allocations and financial metrics used by utilities and other business entities; the transactive incentive signal can instead be formulated to allocate expenses in a way that will induce useful responses for the entity that owns a transactive node. However, a faithful transactive incentive signal formulation should approach the same overall value as for actual expense reporting over long periods of time. There is nothing that would prevent the transactive control and coordination system from supporting markets and revenue accounting in other formulations.

The incentive signal can have a variety of forms or units, but in some embodiments uses units of $/MWhr (or other equivalent, such as a number or value that is proportional (linearly or otherwise) to this unit). Thus, the signal need not be an actual price, but can be representative of a price or economic unit. One tenet of embodiments of the disclosed transactive control scheme is that items that are valued at a location in the system should be combined into one shared signal, and that can be achieved only after there is consensus about a common metric unit to be used by the signal. This principle will help enforce that business entities' operational objectives should fairly compete.

4.4.3 Transactive Load Feedback Signal

Corresponding to a transactive incentive signal time interval is a transactive load feedback signal (e.g., in the kW or other equivalent or representative unit). This transactive feedback signal time series includes the present and future electrical load that is predicted to be supplied through the transactive node during each time interval. In some embodiments, the signal is the sum of the unresponsive electric load that is not affected by the transactive signal and the responsive electric load that can monitor and respond to the transactive incentive signal.

$$\boxed{TFS_{output}(t) = \sum TFS_{unresponsive,input}(t) + \sum TFS_{responsive,input}(t, TIS_{output}(t))}$$

The transactive feedback signal is not a "load forecast" of the type that some utilities prepare as they plan resource commitments. There are no direct penalties to be incurred by subprojects when their transactive feedback signals prove inaccurate. The transactive control approach might diminish the importance of load forecasts in the future if the flexibility provided by transactive control can be shown to displace some of the need for predictive accuracy. Interestingly, the accuracy of a node's transactive feedback signal prediction may always be tested against the true consumption that is measured eventually at the transactive node. In some embodiments, the intelligence at a transactive node can "learn" over time to improve its own predictions. Neighboring transactive nodes learn also from an adjacent transactive node's inaccuracies and may choose to alter or suspect that transactive node's outputs.

In some embodiments, the inputs to the transactive feedback signal at a transactive node include any one or more of the following types of inputs:

- Transactive feedback signals generated from transactive nodes that are immediately downstream;
- Transactive feedback signals generated from smart responsive assets that are controlled from the present transactive node's position in the hierarchy;
- Raw unresponsive load measurements that may be subjected to further computation or modeling to predict the remaining future time intervals; and/or
- Raw responsive load measurements from responsive assets that do not themselves predict and provide transactive feedback signals but instead rely on the transactive node to perform predictions.

4.4.4 Implications for Customer and Utility Incentives

As has been stated, the transactive incentive signal is not intended to account for monetary exchanges or revenue between regional entities. However, the transactive incentive signal could become the foundation for regional exchanges or revenues. The transactive incentive signal may also be used as a basis for customer incentives if the subprojects can establish workable shadow accounts for these customers.

4.5 Transactive Nodes

Any of the physical locations in the electrical connectivity architecture of a power system can be transactive nodes. A node is a location or piece of equipment that electrical power flows through. The term "hierarchy" is used to describe a set of transactive nodes that may extend all the way upstream to bulk generators and all the way downstream to electrical loads.

4.5.1 Responsibilities of a Transactive Node

In certain embodiments, a location or piece of equipment in the electrical connectivity architecture is described as a *transactive node* if it performs one or both of the following:

- Accepts at least one transactive incentive signal time series from upstream and sends a transactive incentive signal time series downstream. If multiple transactive incentive signals are received from upstream, a transactive node blends these incentives into a single transactive incentive signal to be sent downstream.
- Accepts at least one transactive feedback time series from downstream and sends at least one transactive feedback time series upstream. A transactive node can further predict electrical load and can thereby convert raw electrical load meter readings, as necessary, into transactive feedback time series.

A transactive node can also: modify the output transactive incentive signal to address any local operational objective that exists at the transactive node; and/or predict the responsive electric load from any responsive assets that are being controlled from the location of the transactive node.

These responsibilities of a transactive node are summarized by block diagram **1700** of Figure **17**, where the "prediction and control machine" is the intelligence (typically implemented as software executed by computing hardware associated with a transactive node) that modifies the output transactive incentive signal, predicts the behaviors of downstream electrical load, and controls responsive assets at the transactive node.

Any one or more of the following functional behaviors can be carried out by transactive nodes:

- Basic transactive node functions
- Management of electrical constraints
- Management of electrical supply
- Management of responsive assets.

These general functional behaviors help form the basis for a basic building-block model of a transactive node, whose models may be linked together to model the behaviors of the transactive nodes in a completed nodal hierarchy. Each of these functional behaviors is discussed in more detail below.

4.5.2 Basic Transactive Node

This section addresses the most basic functions that a point in the electrical connectivity architecture (hierarchy) performs as part of its role as a transactive node. First, a transactive node desirably is able to receive at least one transactive signal and "blend" the signals into a single transactive signal output to be sent downstream through the hierarchy. For purposes of this discussion, this basic function is termed the incentive blending function and is illustrated in block diagram **1800** in Figure **18**. Secondly, a transactive node desirably is able to receive or meter the downstream electric load that it supplies and aggregate this information and these measurements into a complete transactive feedback signal to be sent upstream through the hierarchy. For purposes of this discussion, this basic function is termed load aggregation, and is also illustrated in Figure **18**.

As a starting point for the design, the default incentive blending function can be assigned as a weighted average of the transactive incentive signals that are received at the transactive node from upstream, where the weighting is performed according to the energy received from each source during the interval. For instance, this weighted average can be formulated as:

$$TSF_{output}(t) = \frac{TFS_{input\,1}(t) \cdot TIS_{input\,1}(t) + TFS_{input\,2}(t) \cdot TIS_{input\,2}(t) + \cdots}{TFS_{input\,1}(t) + TFS_{input\,2}(t) + \cdots}.$$

It is noteworthy that the relative electrical energy to be received from multiple source inputs to a transactive node during a time interval cannot be directly controlled by the transactive node and may only be predicted imperfectly from the transactive node's limited view of the system. This might not be problematic (or even evident) for transactive nodes that exist within largely radial distribution systems, but may become more evident for transactive nodes within highly redundant transmission pathways and near dispatchable generators. This observation results from the more distributed nature of the disclosed transactive control and coordination system and can be contrasted with systems where transmission system conditions are predicted by load flow calculation methods that assume nearly complete system visibility and use simultaneous solution of the entire system's status.

The load aggregation function is conceptually simple, but complexities potentially arise from the breadth of downstream electrical load types and conditions. In principle, the purpose of the load aggregation function is simply to receive or measure electrical load that is supplied through the transactive node and to convert these measurements and this information into the transactive feedback signal, including a prediction of the entire electrical load to be supplied through the transactive node for each time interval. The transactive node can implement this functionality according to one or more of the following cases:

Case 1. If there are transactive nodes immediately downstream from the given transactive node, then the transactive feedback signals that are received from them is already in the right format and should simply be added.

Case 2. The electric load that is not from responsive assets and is not supplied by another downstream node is predicted and converted into the format of the transactive feedback signal. This prediction might rely on an active model of the behaviors of the supplied load or its components. These unresponsive asset behaviors might be influenced by weather, day of week, customer habits, and/or many other conditions, but they are not affected by the transactive incentive signal.

Case 3. A third case is similar to case 2 above but further includes responsiveness to the transactive node's transactive incentive output signal.

4.5.3 Constraint Transactive Node

A transactive node that manages an electrically constrained piece of equipment at the transactive node additionally may modify its output transactive incentive signal to manage this constraint. This additional function is shown in diagram **1900** of Figure **19** in line with the downstream output of the transactive node's transactive incentive signal. This function draws upon predicted electrical load and other local information, including the knowledge of the electrical constraint magnitude.

In summary, the transactive incentive signal can be made responsive to the constraint, and the downstream responsive assets can be made to reduce or curtail their consumption when the transactive incentive signal becomes high.

In contrast to a transactive approach where price is determined by a two-way clearing of a market, embodiments of the disclosed technology base the magnitude of the transactive incentive signal on actual risks and expenses. The transactive incentive signal is therefore not a marginal price but is instead a transparent accumulation of incurred expenses. This approach responds to the criticism received by marginal pricing that it results in more, not less, expense to customers.

If a constraint is to be addressed, the transactive node can be associated with the constrained piece of equipment. This practice can help in situations where it is desirable to have only one output transactive incentive signal be necessary from the perspective of the transactive node.

In some instances, local situation information can also be received from this function, which may generate useful alerts, for example, for system operators. That is, the prediction of constrained operation at a transactive node is reflected in both the transactive incentive and feedback signals at that node, and useful notifications may be generated if thresholds are exceeded in these signals.

4.5.4 Load Transactive Node

This transactive node function addresses a node associated with a load asset and builds on the structure of a basic node. In diagram **2000** of Figure **20** , a function is shown to reside on the path of the output transactive feedback signal. This function allows local situational information to affect prediction of future electric load, but it also includes the effect of the transactive incentive signal toward predicting energy consumption by responsive load. The responsive load is the load consumption of those responsive assets that are controlled at the transactive node. (Responsive assets that are controlled at downstream nodes are also responsive, but their behaviors are already captured in the basic transactive node's summing of signals from downstream transactive nodes.)

Smaller distributed generation can be addressed by using the load transactive node functions. Distributed generators can make their decisions to run or not based on the transactive incentive signal which is provided by the load transactive node functions. When the small generator operates, it effectively reduces downstream electrical load.

The transactive node further uses its version of the transactive incentive signal to functionally control its responsive assets via a toolkit load function selected from a library of such available functions. The output of this function to the responsive assets can depend upon the control method the utility has established for that responsive asset:

- Direct demand response—an event-type of response is initiated by the responsive asset system when the transactive incentive signal exceeds a rather extreme threshold. Events occur infrequently.
- Time–of-use—an event is initiated by the responsive asset system while the transactive incentive signal is within defined boundaries that are exceeded most days. Often used to address system peak load. Includes peak responses where more extreme events are recognized.
- Real-time—a continuum of responses is provided by the responsive asset to the transactive incentive signal. This use case is active most, if not all, days and hours.

These responses are shown conceptually in graph **2100** of Figure **21**. Relative variations in the transactive incentive signal are shown to result in direct demand response, time-of-use (TOU), and real-time response options.

4.5.5 Supply Transactive Node

A supply transactive node function is shown in diagram **2200** of Figure **22** and is similar to a load transactive node function. Both function types attempt to mitigate an imbalance between electrical supply and load, so it is reasonable that their forms would be similar.

This transactive node function is targeted mostly to bulk generation nodes. At these transactive nodes, the base foundation for transactive incentive signals is established. At a supply node, there may be no upstream nodes from which input transactive incentive signals could be received. The function in the path of the output transactive incentive signal is then the initial formulation of the base transactive incentive signal.

Local situational information can be generated or received by this transactive node. The supply transactive node can apply supply control (or a recommendation) if such supply generation is provided at this transactive node. Local information can also be used to inform what fuel expense and other operational expenses should be included into the initial transactive incentive signal at this location.

The incentive signal and the actual expenses of the supply desirably agree over long periods of time, but the function can (while adhering to this stated guideline) address the value of electrical generation in a way that instills useful responses by the region's responsive assets. For example, when this supply transactive node function is applied at wind farms, the created transactive incentive signal can induce the region's responsive assets to consume more of its energy while and near where the wind energy is produced.

4.6 Understanding Generalized Transactive Nodes as Combinations of the Functional Component Types

A set of transactive node functions has been introduced. These functions can be generalized as shown in diagram **2300** of Figure **23**. In particular, diagram **2300** illustrates a single model of a transactive node and its functions. Any one or more aspects of this model can be replicated throughout a transactive control and coordination system to represent a variety of types and instantiations of the system's transactive nodes.

In particular implementations of the transactive system, the output transactive incentive signal becomes an input transactive incentive signal to a transactive node that is immediately downstream; the output transactive feedback signal from a transactive node becomes the input for a transactive node immediately upstream.

4.7 Hierarchy

Block diagram **200** in Figure **2** shows examples of significant transactive node locations that exist within a typical electric power grid. Embodiments of the transactive control technique are unique in that it addresses the power system from bulk generation to end use and back again. Ideally, and in certain embodiments, a complete hierarchy of transactive nodes is defined throughout the power system. In reality, there are parts of the electrical connectivity pathways without transactive nodes. In such cases, some nodes will perform more prediction and do so for more of a distribution system than they would do in a complete hierarchy. Further, in some cases, local constraints and other local operation objectives that might be mitigated by transactive nodes will remain unobserved.

5 Generalized Methods and Systems for Implementing Aspects of the Disclosed Technology.

Having introduced the disclosed technology in the sections, this section presents general methods and systems for performing aspects of the disclosed transactive control approach. The embodiments below should not be construed as limiting and can be performed alone or in combination with any other feature or aspect disclosed herein.

Figure **24** is a flowchart **2400** showing a generalized method for operating a transactive node in a transactive control electrical-energy-allocation system as can be used in any of the disclosed embodiments. The method can be performed using computing hardware (*e.g.*, a computer processor or a specialized integrated circuit). For instance, the method can be performed by computing hardware associated with a transactive node where electrical energy is distributed, generated, and/or consumed.

At **2410**, incentive signal data is computed. The incentive signal data can comprise data indicative of a cost of electric energy at the transactive node at a current time interval and data indicative of a forecasted cost of electric energy at the transactive node at one or more future time intervals. In certain embodiments, the current time interval refers to the imminent (or next-to-occur) interval in which the transactive node will operate.

At **2412**, feedback signal data is computed. The feedback signal data can comprise data indicative of an electric load at the transactive node at the current time interval and data indicative of a forecasted load for electric energy at the transactive node at the one or more future time intervals. In certain embodiments, the current time interval refers to the imminent (or next-to-occur) interval in which the transactive node will operate

At **2414**, the incentive signal data and the feedback signal data is transmitted. For example, the incentive signal data and feedback signal can be transmitted separately or together from one transactive node to each of its neighboring transactive nodes.

In certain embodiments, the data indicative of the cost of electric energy comprises data indicative of a cost of real electrical energy, reactive electrical energy, or a combination of both real and reactive electrical energies at the transactive node at the current time interval. Further, the data indicative of the forecasted cost of electric energy can comprise data indicative of a forecasted cost of real electrical energy, reactive electrical energy, or a combination of both real and reactive electrical energies at the transactive node at the one or more future time intervals. In some embodiments, the data indicative of the electric load comprises data indicative of a real electrical load, reactive electrical load, or a combination of both real and reactive electrical loads at the transactive node at the current time interval. Further, the data indicative of the forecasted load for electric energy can comprise data indicative of a forecasted load of real electrical load, reactive electrical load, or a combination of both real and reactive electrical loads at the transactive node at the one or more future time intervals.

In some embodiments, the incentive signal data further comprises data indicating a confidence level that the data indicative of the cost of electric energy at the transactive node at the current time interval is reliable (e.g., a confidence level for each time interval), and data indicating a confidence level that the data indicative of the forecasted cost of electric energy at the transactive node at the one or more future time intervals is accurate (e.g., a confidence level for each time interval). Further, in certain embodiments, the feedback signal data further comprises data indicating a confidence level that the data indicative of the electric load at the transactive node at the current time interval is accurate, and data indicating a confidence level that the data indicative of the forecasted load for electric energy at the transactive node at the one or more future time intervals is accurate.

In certain embodiments, the method further comprises receiving incentive signal data and feedback signal data from one or more neighboring transactive nodes. In such embodiments, the computation of the incentive signal data can be based at least in part on the received incentive signal data, and/or the computation of the feedback signal data can be based at least in part on the received feedback signal data.

Figure **25** is a flowchart **2500** showing another generalized method for operating a transactive node in a transactive control electrical-energy-allocation system as can be used in any of the disclosed embodiments. The method can be performed using computing hardware (*e.g.*, a computer processor or a specialized integrated circuit). For instance, the method can be performed by computing hardware associated with a transactive node where electrical energy is distributed, generated, and/or consumed.

At **2510**, incentive signal data is received at the transactive node from two or more neighboring transactive nodes. The incentive signal data from the two or more neighboring transactive nodes can comprise data indicative of at least a cost of electric energy at a current time interval. In certain embodiments, the incentive signal data comprises data indicative of the cost of electric energy at the current time interval (e.g., the delivered unit cost of the energy at that node) and data indicative of a forecasted cost of electric energy at one or more future time intervals. In certain embodiments, the current time interval refers to the imminent (or next-to-occur) interval in which the transactive node will operate

At **2512**, aggregated incentive signal data is computed based at least in part on the incentive signal data from the two or more neighboring transactive nodes. In some embodiments, the aggregated incentive signal data comprises data indicative of the aggregated cost of electric energy at the current time interval and data indicative of a forecasted aggregated cost of electric energy at one or more future time intervals. Further, in some embodiments, the aggregated incentive signal data comprises a weighted sum of the incentive signal data from the two or more neighboring transactive nodes. In certain embodiments, the aggregated incentive signal data is further modified to provide an incentive or disincentive to the further transactive node based on local conditions at the transactive node. In certain embodiments, the current time interval refers to the imminent (or next-to-occur) interval in which the transactive node will operate

At **2514**, the aggregated incentive signal data is transmitted to a further transactive node (e.g., a neighboring transactive node).

In some embodiments, the received incentive signal data and the transmitted aggregated incentive signal data comprise data indicative of a cost of real electrical energy, reactive electrical energy, or a combination of both real and reactive electrical energies. In certain embodiments, the received incentive signal data further includes data indicating a confidence level of the received incentive signal data (e.g., a confidence level for each time interval). And in some embodiments, the transmitted incentive signal data further includes data indicating a confidence level of the transmitted incentive signal data (e.g., a confidence level for each time interval).

In some embodiments, the method further comprises receiving feedback signal data at the transactive node from the two or more neighboring transactive nodes, the feedback signal data from the two or more neighboring transactive nodes comprising data indicative of at least an electric load for electric energy at a current time interval; computing aggregated feedback signal data based at least in part on the feedback signal data from the two or more neighboring transactive nodes; and transmitting the aggregated feedback signal data to the further transactive node. In such embodiments, the received feedback signal data can comprise data indicative of the electric load for electric energy at the current time interval and data indicative of a forecasted load of electric energy at the one or more future time intervals, and the aggregated feedback signal data can comprise data indicative of the aggregated load of electric energy at the current time interval and data indicative of a forecasted aggregated load of electric energy at one or more future time intervals.

Figure **26** is a flowchart **2600** showing another generalized method for operating a transactive node in a transactive control electrical-energy-allocation system as can be used in any of the disclosed embodiments. The method can be performed using computing hardware (*e.g.*, a computer processor or a specialized integrated circuit). For instance, the method can be performed by computing hardware associated with a transactive node where electrical energy is distributed, generated, and/or consumed.

At **2610**, feedback signal data is received at a transactive node from two or more neighboring transactive nodes. The feedback signal data from the two or more neighboring transactive nodes can comprise data indicative of at least an electric load for electric energy at a current time interval. In certain embodiments, the received feedback signal data comprises data indicative of the electric load of electric energy at the current time interval and data indicative of a forecasted load of electric energy at one or more future time intervals. In certain embodiments, the current time interval refers to the imminent (or next-to-occur) interval in which the transactive node will operate

At **2612**, aggregated feedback signal data is computed based at least in part on the feedback signal data from the two or more neighboring transactive nodes. In certain embodiments, the aggregated feedback signal data comprises data indicative of the aggregated load of electric energy at the current time interval and data indicative of a forecasted aggregated load of electric energy at the one or more future time intervals.

At **2614**, the aggregated feedback signal data is transmitted to a further transactive node.

In certain embodiments, the received feedback signal data and the transmitted aggregated feedback signal data comprise data indicative of a real electrical load, reactive electrical load, or a combination of both real and reactive electrical loads. In some embodiments, the received feedback signal data further includes data indicating a confidence level of the received feedback signal data (e.g., a confidence level for each time interval). And in certain embodiments, the transmitted feedback signal data further includes data indicating a confidence level of the transmitted feedback signal data (e.g., a confidence level for each time interval).

In some embodiments, the method further comprises receiving incentive signal data at the transactive node from the two or more neighboring transactive nodes, the incentive signal data from the two or more neighboring transactive nodes comprising data indicative of at least a cost of electric energy at the current time interval; computing aggregated incentive signal data based at least in part on the incentive signal data from the two or more neighboring transactive nodes; and transmitting the aggregated incentive signal data to the further transactive node. In such embodiments, the received incentive signal data can comprise data indicative of the cost of electric energy at the current time interval and data indicative of a forecasted cost of electric energy at the one or more future time intervals, and the aggregated incentive signal data can comprise data indicative of the aggregated cost of electric energy at the current time interval and data indicative of a forecasted aggregated cost of electric energy at one or more future time intervals.

Figure **27** is a flowchart **2700** showing another generalized method for operating a transactive node in a transactive control electrical-energy-allocation system as can be used in any of the disclosed embodiments. The method can be performed using computing hardware (*e.g.*, a computer processor or a specialized integrated circuit). For instance, the method can be performed by computing hardware associated with a transactive node where electrical energy is distributed, generated, and/or consumed. The method can be performed for a transactive node associated with one or more electric resources, one or more electric loads, or a combination of both electric resources and loads.

At **2710**, one or more functions from a library of functions are selected. The selection can be based at least in part on the type of one or more electric resources or electric loads associated with the transactive node. In certain embodiments, the selected one or more functions are adapted for the type of electrical load or electrical supply associated with the transactive node. In some embodiments, the configuring comprises causing computing hardware used to implement the transactive node to execute a software program for performing computations using the selected one or more functions. In certain embodiments, the selected one or more functions include a function that computes data representing one or more of energy, an energy cost, or an incentive for one or more electric resources associated with the transactive node. In some embodiments, the selected one or more functions include a function that computes data representing one or more of a predicted inelastic load or changes in elastic load for one or more electric loads associated with the transactive node

At **2712**, the transactive node is configured to compute transactive signals using the selected one or more functions.

In some embodiments, the method can comprise accessing a database storing the library of functions (e.g., a locally stored database or a database remotely located from the transactive node).

Further, the library of functions can be an extensible library. For example, the library can be expanded to include newly formulated functions. Further, in some implementations, existing functions may be selected from the library, edited by a relevant party (e.g., a utility or system administrator), and returned to the library as a newly available function with modified features and capabilities. The parties that have access to editing and adding library functions can vary from implementation to implementation, and can encompass a wide variety of parties involved in the power transmission infrastructure. In some instances, the parties who can edit and/or add functions is limited to some selected group (e.g., system regulators or to a single utility).

Also disclosed herein are several embodiments for systems for distributing electricity. One of the disclosed systems is a system for distributing electricity, comprising: a plurality of transactive nodes, each transactive node being associated with one or more electric resources, one or more electric loads, or a combination of one or more electric resources and loads; and a network connected to the transactive nodes to facilitate communication between the transactive nodes. In these embodiments, the transactive nodes are configured to exchange incentive and feedback signals with one another in order to determine an electrical demand in the system for a current time interval and to provide an electrical supply sufficient to meet the electrical demand for the current time interval. In certain embodiments, the current time interval refers to the imminent (or next-to-occur) interval in which the transactive nodes will operate

In certain embodiments, the transactive nodes are further configured to exchange incentive and feedback signals for two or more future time intervals in addition to the incentive and feedback signals for the current time interval. In some embodiments, the two or more future time intervals have increasingly coarser granularity. In certain embodiments, at least one of the transactive nodes modifies one or both of its incentive or feedback signals in response to previously received incentive and feedback signals. In some embodiments, the at least one of the transactive nodes is associated with an elastic load, and wherein the modified incentive or feedback signals corresponds to a predicted change in the elastic load. In certain embodiments, the at least one of the transactive nodes is associated with an electrical resource, and the modifed incentive or feedback signals corresponds to a change in the electrical resource. In further embodiments, the at least one of the transactive nodes is associated with an electrical resource, and the modified incentive signals correspond to a change in local conditions.

In certain embodiments, one or more of the transactive nodes compute their respective incentive and feedback signals using functions selected from a library of functions. Still further, in some embodiments, the incentive and feedback signals further include confidence level data indicating a respective reliability of the incentive and feedback signals.

Another system disclosed herein is a system for distributing electricity, comprising: a plurality of transactive nodes, each transactive node being associated with one or more electric resources, one or more electric loads, or a combination of one or more electric resources and loads; and a network connected to the transactive nodes and facilitating communication between the transactive nodes. In these embodiments, the transactive nodes are configured to exchange sets of signals with one another in order to determine an electrical demand in the system for a current time interval and to provide an electrical supply sufficient to meet the electrical demand for the current time interval. Each set of signals includes signals for determining the electric loads and supplies for the current time interval as well as signals for determining the electric loads and supplies for two or more future time intervals. In certain embodiments, the current time interval refers to the imminent (or next-to-occur) interval in which the transactive nodes will operate

In some embodiments, the future time intervals have increasingly longer durations as the time intervals are farther into the future relative to the current time interval. In other embodiments, the transactive nodes are configured to update the values of the sets of signals at an update frequency, the update frequency corresponding to a duration of the current time interval. In some embodiments, the transactive nodes are configured to exchange the set of signals with one another iteratively over time such that the signals for a respective time interval stabilize as the respective time interval approaches the current time interval.

In certain embodiments, the transactive nodes are configured to exchange the set of signals with one another on an asynchronous event-driven basis or a clock-driven basis. In some embodiments, a respective set of the transactive nodes are configured to iteratively exchange a set of signals with one another until the exchanged set of signals converges to within an acceptable degree of tolerance. In certain embodiments, a transactive node in the respective set of the transactive nodes is further configured to transmit an updated set of signals when local conditions at the transactive node cause the updated set of signals to deviate from a previously transmitted set of signals beyond a relaxation criterion. In some embodiments, the sets of signals further include confidence level data indicating a respective reliability of the exchanged signals (e.g., a confidence level for each time interval).

Another system disclosed herein is a system for distributing electricity, comprising: a plurality of transactive nodes, each transactive node being associated with one or more electric supplies, one or more electric loads, or a combination of one or more electric supplies and loads; and a network connected to the transactive nodes and facilitating communication between the transactive nodes. In these embodiments, the transactive nodes are configured to exchange sets of signals with one another in order to determine an electrical demand in the system for a current time interval and to provide an electrical supply sufficient to meet the electrical demand for the current time interval, a respective one of the transactive nodes being configured to compute its incentive and feedback signals using one or more functions selected from a library of functions. In certain embodiments, the current time interval refers to the imminent (or next-to-occur) interval in which the transactive nodes will operate

In certain embodiments, the one or more functions selected from the library of functions are selected based on the type and number of electrical supplies and electrical loads with which the respective transactive node is associated. The one or more functions can be selected from a group of resource functions comprising one or more of: (a) a resource function adapted to account for imported electrical energy, (b) a resource function adapted to account for a renewable energy resource, (c) a resource function adapted to account for fossil fuel generation, (d) a resource function adapted to account for general infrastructure cost, (e) a resource function adapted to account for system constraints, (f) a resource function adapted to account for system energy losses, (g) a resource function adapted to account for demand charges, and (h) a resource function adapted to account for market impacts. The one or more functions can also be selected from a group of load functions comprising one or more of: (a) a load function adapted to account for a bulk inelastic load, (b) a load function adapted to account for an event-driven demand response, (c) a load function adapted to account for a time-of-use demand response, and (d) a load function adapted to account for a real-time continuum demand response.

In some embodiments, the respective one of the transactive nodes controls one or more elastic loads and adjusts the one or more elastic loads in response to the feedback and incentive signals received at the respective one of the transactive nodes. In certain embodiments, the one or more functions are implemented by individual software modules that can be combined with one another to implement the desired transactor behavior for the respective one of the transactive nodes.

In certain embodiments, through the use of the one or more functions, the respective one of the transactive nodes computes a control signal selected from a set of signed whole numbers and communicates the computed control signal to one or more loads, resources, or loads and resources associated with the respective one of the transactive nodes. The computed control signal can be interpreted by an electrical generator or set of electrical generators as a fraction of the generator's or generators' rated generation capacity. The computed control signal is interpreted by an electrical load or set of electrical laods as a fraction of the load's or loads' rated power.

It should be understood that in embodiments of the disclosed technology, a transactive node may host multiple toolkit funcitons, including any combination of multiple resource and incentive functions, multiple load functions, or combinations of both resource and incentive and load functions. For instance, the resource and/or incentive functions used at a transactive node will typically depend on the location of the transactive node in a power grid topology, and on the one or more resources and/or loads for which the transactive node is responsible. This ability to "mix and match" resource and incentive functions while still maintaining a common transactive signal communication structure gives embodiments of the disclosed technology wide flexibility and scalability for implementing a transactive control system.

6 Further Details and Embodiments

Having introduced the disclosed technology, this section includes four supplemental Appendices that provide additional details and configurations that can be used in implementations of the technology. The specific implementations disclosed below should not be construed as limiting. Further, any one or more of the features or aspects disclosed below can be used alone or in conjunction with any other feature or aspect of the disclosed technology discussed herein. Some portions of the appendices may, in some instances, be repetitive to other portions of this application, but such portions are included for the sake of completeness.

6.1 Appendix A – Transactive State Model

6.1.1 Purpose

A transactive control and coordination system is a network of loosely connected, interacting transactive nodes. This appendix states a high-level state model for a transactive node and types of connections that a transactive node desirably manages. This appendix should provide valuable guidance to system designers who are implementing a transactive control and coordination system from the perspective of a transactive node.

This appendix defines and discusses

- example attributes of a transactive node and four example types of connections
- the organization of these attributes into groups—transactive node, general connection, transactive neighbor, system manager, asset, and local information
- example allowed states within the high-level transactive node state model
- example functions and events by which attributes become changed and by which the states are navigated in this state model

- example state transition tables and diagrams for the respective transactive node and its connections..

6.1.2 Structure

In some embodiments, a transactive node manages its own set of attributes and additionally manages additional types of connection. In certain implementations the transactive node manages four types of connections—connections to transactive neighbors, system managers, assets, and local input information. All four connection types can share a set of connection attributes in common in order to manage connections between this transactive node and each transactive neighbor, system manager, asset, or local input information. An example of this structure has been laid out in diagram **2800** in Figure **28**.

6.1.3 Transactive Node States and State Diagram

In certain embodiments, a transactive node has five states available to it as shown in the state transition diagram **2900** of Figure **29**:

> *1 - New or Terminated*—initial and terminal state where the transactive node attributes are not defined. The transactive node leaves and returns to this state by running or terminating an executable program.
>
> *2 – Under Local Control*—intermediate state where the transactive node executable process is up and running, but the transactive node and its connections are not adequately configured. Few, if any, of this transactive node's connections have been completed between this transactive node and its transactive neighbors, system managers, assets, or local information sources, which collectively will be referred to as the transactive node's "connection partners." A transactive node enters this state when a transactive node executable program is run or when a *Configuration Test* fails.
>
> *3 – Configured*—intermediate state where certain transactive node attributes (those transactive node attributes having asterisks in Figure **28**) have been defined and each of the connections that this transactive node manages is also in its *Configured* state. A transactive node enters this state by passing a *Configuration Test* or by failing a *Connection Test*.
>
> *4 – Connected & Configured*—a transactive node state that has been *Configured* and now each of the connections that this transactive node manages is in its *Connected* (or

temporarily in its *Lost Connection*) state. A transactive node enters this state by passing a *Connection Test*, by receiving and accepting a *Halt Operations* command, or by encountering a *Fatal Operational Event.*

***5** – Operational*—a transactive node that has been *Connected* & *Configured* and which now interacts with its connection partners according to its algorithmic responsibilities of membership in a transactive control and coordination system. The algorithmic responsibilities are addressed elsewhere as a "toolkit framework" of computational algorithms and a suite of "toolkit library functions" that may be incorporated to represent the more unique and individual algorithmic responsibilities of transactive nodes. The toolkit framework and the toolkit library functions are described in more detail in Appendices B and C. A transactive node enters this state by receiving and accepting an *Operate* command.

The identifying numbers that have been applied to the functions and events in Figure **29** are derived from the prior and end states. A letter is appended wherever multiple functions or events achieve the same state transition. For example, the function numbered "**54**b" (e.g., a *Halt Operations* command in Figure **29**), is the second state transition that has been defined from state "**5**" to state "**4**." These same function and event numbers will be used in corresponding state transition table.

6.1.4 Connection States and State Diagram

Each connection has four allowed states as shown in diagram **3000** of Figure **30**. The only details that really change between the four types of connections are those attributes that are tested if a *Connection Configuration Test* is to be passed for a given connection. These are the connection states and their descriptions:

***1** – Listed*—a connection has been *listed* when its identifier appears among those in any of these corresponding connection attribute lists:

- ***49*** - *List of Transactive Neighbors* (a transactive node attribute)
- ***50*** - *List of System Managers* (a transactive node attribute)
- ***51*** - *List of Assets* (a transactive node attribute)
- ***38*** - *List of Local Information Connections* (an asset connection attribute)

There is no expectation that any of the corresponding attributes have been configured in this state. A connection reaches this state by becoming listed in one of the attributes above, which may occur as a transactive node executable program is being run or thereafter using the *Configure* command. This is an initial and terminal state of any connection.

2 – Configured—certain attributes (see asterisks in Figure **28**) of this connection have been configured and are not empty. This connection enters this state by passing a *Connection Configuration Test*, by accepting a *Disconnect* command that has been directed to this connection, or when a Terminate *Connection Event* occurs after this connection has been in its *Lost Connection* state, which event indicates that either a timeout duration has expired or that too many *Loss of Connection Events* have occurred in the past hour or day. This is an intermediate state.

3 – Connected—a communication link (a "connection") has become successfully established between this transactive node and one of its connection partners via this connection. A connection enters this state by receiving and accepting a *Connect* command or by having the connection re-established from a *Lost Connection* state by a *Connect* command or a *Re-Establish Connection Event*. This is an intermediate state, but a connection should be expected to remain in this state most of the time.

4 – Lost Connection—the state of a connection while the connection between this transactive node and one of its connection partners via this connection has become broken or severed. This temporary intermediate state may be entered by a connection only by a *Loss of Connection Event*. The connection should thereafter be either re-established by a *Connect* command or *Re-Establish Connection Event*, or the connection should become disconnected by a *Disconnect* command or by a *Terminate Connection Event*.

Again, the identifying numbers and letters that prepend the functions and events in Figure **30** are derived from prior and end states and will be used also to identify these same transitions in state transition tables.

6.1.5 The Meaning of Attribute Dictionary Columns

Table 7 is a dictionary of example attributes that can be used to define the state of a transactive node. Later in this appendix, attribute dictionaries will be presented to address attributes of the four types of connections. The meanings of the columns in these dictionary tables are as follows.

- Attribute—structured list of attributes (properties, characteristics) defines the pertinent properties of a class of objects. Assigning specific values to the full set of attributes, creates a specific instance or member of the class. Grouping certain attributes into subsets defines the states of an object, including a single start state, one or more intermediate states, and one or more final states.
- Attribute Name—a string of alphanumeric (alphabetic, numeric) and possibly special characters given to the attribute for reference.
- Description—an easy-to-read narrative about the attribute, clearly distinguishing it from other attributes.
- Role—the reason the attribute is important for: **1**) the definition of an object, and **2**) the application of an attribute in the process that directs actions to instantiate a specific object.
- Type—the attribute may represent a type of number, character string, a pointer to a procedure, set of algorithms, names of other classes, an address, or an array of types.
- Format—the specific arrangement of the characters or the parts of the assigned attribute value(s).
- Range of Values—the specific set of values a process may assign to an attribute, such as least value and greatest value for numbers.
- Security—the level of security assigned to an attribute, the identification of the entities (people, systems) authorized to access an attribute, and whether the entities have the right only to read the value of the attribute or to both read and write the attribute value(s).

6.1.6 Transactive Node Attribute Dictionary

The transactive node attribute group contains those attributes that stand alone and refer to one transactive node and its transactive node state model. An example attribute dictionary is shown in Table **7**.

Table **8** that follows is a summary of which of these attributes can be added, checked, or modified by the set of commands and events that occur within the state transition table (Table **7**), as were introduced in the state transition diagram **2900** of Figure **29**.

Table 7: Dictionary of the Transactive Node Attributes

No.	Attribute Name	Description	Role	Type	Format	Range of values
1	*Node ID*	Unique ID of this transactive node.	This transactive node's name that may be used to refer to it. This is a attribute that is desirably found to have been configured during *Configuration Tests.*	Character string	**0-9**, A-Z Example: "UT-**01**"	See topology for the transactive control and coordination system where transmission zone, balancing authority, utility, and site names have been stated.
3	*Node Type*	The type of transactive control node. Four types have been identified: Transmission Zone (TZ) Balancing Authority (BA), Utility (UT), Site		Character string		TZ, BA, UT, ST

No.	Attribute Name	Description	Role	Type	Format	Range of values
		(ST)				
4	*Geographical Location of Node*	The representative physical location of this transactive node.	Perhaps useful for future global information system (GIS) representations.		(latitude, longitude)	(-**90** to **90**, **0** – **360**) degrees (-pi to pi, **0** – **2***pi) radians Default value: (null,null)
5	*Node Version**	The implementation version for the instantiated transactive node at the time the *Run Node Executable* command is issued. This executable file represents a "version" for the transactive node overall.	To keep track of successions of software during incremental improvements, troubleshooting, testing. This is an attribute that is desirably found to have been configured during *Configuration Tests*.	Two alphanumeric items	"Filename, **##.##**", where **##.##** are the major and minor version numbers of this file, respectively.	"Filename" should be an allowable executable filename. "**##.##**" major and minor versions anticipated from "**0.00**" to "**99.99**".
7	*Node Status**	The state of this transactive node within this state model.	Unambiguous representation of the state of this transactive node within this state model. This is an attribute that is desirably found to	Single integer	Example: "**1**"	"**1**" – New or Terminated, "**2**" – Under Local Control, "**3**" – Configured, "**4**" – Connected & Configured, "**5**" - Operational

No.	Attribute Name	Description	Role	Type	Format	Range of values
			have been configured during *Configuration Tests.*			
8	*Mode*	The current mode of operation.		Single character string		"Experimental", "Production", "Test"
9	*Update Frequency* *	The frequency used to update *TIS* and *TFS*. Units are "updates per hour".	The update frequency may change between testing and operation. This is an attribute that is desirably found to have been configured during *Configuration Tests.*	Single integer	Integer Example: "**12**"	From **1** to **3600**. The Demonstration will most often use "**12**", meaning one update is performed every **5** minutes.
16	*Electrical Topology Location*	The logical location of a transactive node in an electrical system		Character string	Varied	Varied
18	*Time**	Present time in UTC Format. Time is coordinated across the system of transactive nodes to within **500** millisecond	Time is used to mark when node state transitions occur and also to support timing of events related to **9** - *Update Frequency*.	See UTC standard.	See UTC standard	

No.	Attribute Name	Description	Role	Type	Format	Range of values
		s, or so.	Each transactive node calculates transactive signal interval start times starting from this, the present time. This is an attribute that is desirably found to have been configured during *Configuration Tests.*			
21	*Processing Time Delay*	The time delay for this node within the processing time interval for the system of transactive control nodes	The time delay is used to manage the time sequence relationships	Varied	Varied	Varied
22	*Time Out*	The time to wait for receipt of TIS and TFS from adjacent nodes	If expected TIS / TFS are not received before the time out then the node proceeds with available information and reports an associated change in	Varied	Varied	Varied

No.	Attribute Name	Description	Role	Type	Format	Range of values
			data quality values			
34	*Resource Schedules and Cost Buffer*	A storage location described in the toolkit framework. Records of this storage location possess information about resources and incentives, most of which are being applied via toolkit functions.	See toolkit framework. This storage location has data that is relevant to the formulation of both the TIS and TFS.	List of series. The individual records will probably resemble TIS and TFS. See toolkit framework.	Expected to be very similar to TIS and TFS. See the toolkit framework.	Reasonable ranges may be asserted.
38	*Current IST Series Buffer*	The series of interval start times (IST) that have been calculated and will be used to define the intervals of transactive signals that are being formulated.	A storage location described in the toolkit framework. An interim data storage location within the toolkit framework. Refer to the toolkit framework.	One series of times using UTC standard. See toolkit framework.	See the toolkit framework. Series of times in UTC standard format.	See toolkit framework. The Demonstration has defined **56** intervals. The intervals can align with one of the **12** major division of an hour.
39	*Input Transactive Signals Buffer*	A storage location described in the toolkit framework. Records include at least the most	Holding place for most recent transactive signals that have been received. Holds at least	List of TIS and TFS (e.g., a list of series). See toolkit framework.	See transactive signal formats and XML schema.	Refer to range attributes of TIS and TFS.

No.	Attribute Name	Description	Role	Type	Format	Range of values
		recently received transactive signals.	attributes ***23** - Receive TIS Buffer* and ***24** - Receive TFS Buffer*, but may also retain records of prior examples. An interim data storage location within the toolkit framework. Refer to the toolkit framework.			
40	*Resource and Incentive Input Buffer*	A storage location described in the toolkit framework.	Place where the input "other local conditions" that will be invoked by resource and incentive toolkit functions should be held and managed. Attribute ***25** – Local Information Source* states the sources that should supply the contents of this storage location. An interim	List of various items and series data as should be defined for each toolkit resource and incentive function. Refer to toolkit resource and incentive functions that are used at this transactive node where these specificatio	Various. See the toolkit framework. See individual toolkit resource and incentive functions, where the contents and formats should be specified.	Various for records that should be defined in toolkit functions.

No.	Attribute Name	Description	Role	Type	Format	Range of values
			data storage location within the toolkit framework. Refer to the toolkit framework.	ns should be made.		
41	*Load Function Input Buffer*	A storage location described in the toolkit framework.	Place where the input "other local conditions" that will be invoked by load toolkit functions should be held and managed. Attribute ***25*** - *Other Local Conditions Source* states the sources that should supply the contents of this storage location. An interim data storage location within the toolkit framework. Refer to the toolkit framework.	List of various items and series data as should be defined for each toolkit load function. Refer to toolkit load functions that are used at this transactive node where these specifications should be made.	Various. See the toolkit framework. See individual toolkit load functions, where the contents and formats should be specified.	Various for records that should be defined in toolkit functions.
42	*Output TIS Buffer*	A storage location described in the toolkit framework.	The formulated TIS is held here and may be	One TIS	See TIS	See range attributes of TIS

No.	Attribute Name	Description	Role	Type	Format	Range of values
		Place where updated TIS is held until it can be distributed.	replaced and further updated until it is finally distributed to transactive neighbors (and maybe other entities). See attribute ***12*** *- Send TIS Targets.* An interim data storage location within the toolkit framework. Refer to the toolkit framework.			
43	*Output TFS Buffer*	A storage location described in the toolkit framework. Place where updated TIS is held until it can be distributed.	The formulated TFS is held here and may be replaced and further updated until it is finally distributed to transactive neighbors (and maybe other entities). See attribute ***13*** *- Send TFS Targets.* An interim data storage location within the toolkit framework.	One TFS.	See TFS	See range attributes of TFS

No.	Attribute Name	Description	Role	Type	Format	Range of values
			Refer to the toolkit framework.			
44	*Total Predicted Resource Buffer*	A storage location described in the toolkit framework.	Sum of average power that is generated within or imported into a transactive node during future intervals. Compared against total load during the formulation of TFS series. An interim data storage location within the toolkit framework. Refer to the toolkit framework.	List of series. Contents should be similar to TFS with same format.	Modeled after, or identical to, a TFS format.	Represents total average generated power and imported power during an interval. Reasonable ranges can be stated, but there is no such test in the present model.
45	*Inelastic Load Prediction Buffer*	A storage location described in the toolkit framework.	Records are the inelastic load predicted from one toolkit load function for future intervals. Used to predict total load at future intervals. An interim data storage location	List of series. Contents should be similar to TFS with same format.	Modeled after, or identical to, a TFS format.	Records of this list represent the load being modeled by a toolkit load function. Reasonable ranges can be stated, but there is no such test in the present model.

No.	Attribute Name	Description	Role	Type	Format	Range of values
			within the toolkit framework. Refer to the toolkit framework.			
46	*Elastic Load Prediction Buffer*	A storage location described in the toolkit framework.	Records are the changes to elastic load that are predicted from one toolkit load function for future intervals. Used to predict total load at future intervals. An interim data storage location within the toolkit framework. Refer to the toolkit framework.	List of series. Contents should be similar to TFS with same format.	Modeled after, or identical to, a TFS format.	Records of this list represent the change in elastic component of a load that is being modeled by a toolkit load function. Reasonable ranges can be stated, but there is no such test in the present model.
47	*Predicted Inelastic and Elastic Load Buffer*	A storage location described in the toolkit framework. An interim data storage location within the toolkit framework. Refer to the toolkit framework.	An interim data storage location within the toolkit framework. Refer to the toolkit framework. An interim data storage location within the toolkit framework. Refer to the	List of series. Contents should be similar to TFS with same format.	Modeled after, or identical to, a TFS format.	Records of this list represent total load of a transactive node. Reasonable ranges can be stated, but there is no such test in the present model.

No.	Attribute Name	Description	Role	Type	Format	Range of values
			toolkit framework.			
49	*List of Transactive Neighbors*	List of transactive nodes with which this transactive node exchanges electrical energy and will therefore exchange transactive signals.	This transactive node declares transactive neighbors that it plans to interact with. A transactive neighbor that appears on this list is eligible to enter its *Listed* state after its **52** – *Transactive Neighbor ID* has become configured. This attribute is checked during *Configuration Tests* and *Connection Tests* to see if expected transactive neighbors have become *Configured* and *Connected*.	Comma-separated list of character strings	Example **#1**: "UT**06**", which is the Demonstration's identifier for an demonstration utility.	See system topology. List should include nearby transactive nodes with which this transactive node expects to exchange energy. Naming practice should be the same here and for attribute ***52*** – *Transactive Neighbor ID*, a Transactive Neighbor attribute.
50	*List of System Managers*	List of entities of a transactive control and coordination system that will be granted at	This is the attribute by which this transactive node declares which entities it will	Comma-separated list of character strings	Example **#1** : "EI**01**" to represent the system manager, from which system managemen	See system topology. Naming practice should be the same here and for

No.	Attribute Name	Description	Role	Type	Format	Range of values
		least limited permission to make system management commands to this transactive node. A system manager may be, but is not necessarily, also a transactive neighbor.	allow to make system management commands. The system managers instantiate a connection, and this transactive node accepts a responsibility to maintain the connection to each system manager. A system manager in this list is eligible to enter its *Listed* state. For each *Listed* system manager, this transactive node should manage and monitor its state to enter either the **3** - *Configured* or **4** - *Configured & Connected* transactive node states, and for which *Configuration Tests* and		t command will likely be received. Example #2: "**UT06**", which is the Demonstration's identifier for a demonstration utility, which may be both a system manager to the transactive nodes that it owns and a transactive neighbor, too.	attribute **53** - *System Manager ID*, a System Manager attribute.

No.	Attribute Name	Description	Role	Type	Format	Range of values
			Connection Tests are conducted.			

No.	Attribute Name	Description	Role	Type	Format	Range of values
51	*List of Assets*	List of generation resources, incentives, and loads that are engaged and used by this transactive node.	This is the attribute in which a transactive node declares its assets. Each asset should be accompanied by a toolkit function that defines its predicted participation in ways that affect transactive signals that are formulated at this transactive node. The assets listed here are eligible to enter their Listed states after attribute **2** – *Asset ID* has been configured. This attribute is checked during *Configuration Tests* and *Connection Tests* to see if expected assets have become *Configured* and *Connected*.	Comma-separated list of character strings	Example **#1** : "AV**01**" to represent an asset system of Avista Utilities.	See toolkit framework. See respective toolkit function for a given asset. Naming practice should be the same here and for attribute **2** – *Asset ID*, an Asset attribute.

No.	Attribute Name	Description	Role	Type	Format	Range of values
57	*Interval Durations**	An ordered list of interval durations in minutes that will be used by this transactive node as it formulates its transactive signals. Order is from first to most distant into the future.	This attribute along with ***58*** – *Numbers of Intervals* states the durations of the intervals that this transactive node will represent in each of the transactive signals that it calculates. The number of series elements in this attribute and in ***58*** – *Numbers of Intervals* should be identical at the times transactive signals are being calculated. This attribute creates no expectation that transactive neighbors will have used the same interval durations. This transactive node should be quite flexible in its ability to	Comma-separated list of integers that represent interval durations in minutes	Demonstration example: {**5**, **15**, **60**, **360**, **1440**}, representing **5** minutes, **15** minutes, **1** hour, **6** hours, and **1** day. The **1**-day intervals are most distant into the future. In the above example, the last sample of each duration has a flexible duration that may vary between the present and the following durations. This is done to keep intervals aligned with hourly market data.	Integer values between **1** and **1440**. An allowed number of series elements may be specified in the future but will not be an issue for the Demonstration that will use only **5** different interval durations. Note that this approach that uses integer minutes will limit the practice of intervals that are shorter than **1** minute in the future. The number of series elements in this attribute and in ***58*** – *Numbers of Intervals* should be identical at the times transactive signals are being calculated.

No.	Attribute Name	Description	Role	Type	Format	Range of values
			receive and interpret diverse time series information.			

No.	Attribute Name	Description	Role	Type	Format	Range of values
58	*Numbers of Intervals**	An ordered list of the number of each of the **57** - *Interval Durations* that will be used by this transactive node as it formulates its transactive signals. Order is from first to most distant into the future.	This attribute along with ***57*** – *Interval Durations* states the number of the intervals of each duration that this transactive node will represent in each of the transactive signals that it calculates. The number of series elements in this attribute and in ***57*** – *Interval Durations* should be identical at the times transactive signals are being calculated. This attribute creates no expectation that transactive neighbors will have used the same intervals. This transactive node should be quite flexible in its ability to	Comma-separated list of integers that represent the number of each corresponding interval duration that is listed in ***57*** – *Interval Durations*.	Demonstration example: {**12**, **20**, **18**, **4**, **2**}, representing that there will be **12** **5**-minute, **20** **15**-minute, **18** **1**-hour, **4** **6**-hour, and **2** **1**-day intervals. The last member of each interval duration (e.g., the **12**th, **32**nd, **50**th, and **54**th intervals) varies in duration between the durations of the present and next intervals.	No explicit limit has been placed on the magnitude of each element; however, an element would unlikely be greater than **10,080**—the number of minutes in a week. An allowed number of series elements may be specified in the future but will not be an issue for the Demonstration that will use only **5** different interval durations. The number of series elements in this attribute and in ***57*** – *Interval Durations* should be identical at the times transactive signals are being calculated.

No.	Attribute Name	Description	Role	Type	Format	Range of values
			receive and interpret diverse time series information.			

Table 8. Ways in Which Transactive Node Attributes may be affected by this State Model's Commands and Events

Attribute #	Attribute Name	Run Node Executable Command	Configure Command	Operate Command	Halt Operations Command	Terminate Node Command	Configuration Test**	Connection Test**	Handle Fatal Operational Event	Handle Non-Fatal Operational Event

Attribute #	**Attribute Name**	**Run Node Executable Command**	**Configure Command**	**Operate Command**	**Halt Operations Command**	**Terminate Node Command**	**Configuration Test****	**Connection Test****	**Handle Fatal Operational Event**	**Handle Non-Fatal Operational Event**
1	*Node ID**	++				-	C			
5	*Node Version**	++				-	C			
7	*Node Status**	(C)++	C0	C0	C0	C-	C0	C0	C00	C
9	*Update Frequency**	+	+0			-	C		(C)	(C)
18	*Time**	+	+0			-	C		(C)	(C)
57	*Interval Durations**	+	+0			-	C			
58	*Numbers of Intervals**	+	+0			-	C			
49	*List of Transactive Neighbors*	+	+0			-	C	C		
50	*List of System Managers*	+	+0			-	C	C		
51	*List of Assets*	+	+0			-	C	C		
34	*Resource Schedules and Cost Buffer*	+	+0			-				
38	*Current IST Series Buffer*	+	+0			-				
39	*Input Transactive Signals Buffer*	+	+0			-				
40	*Resource and Incentive Input Buffer*	+	+0			-				
41	*Load Function Input Buffer*	+	+0			-				
42	*Output TIS Buffer*	+	+0			-				
43	*Output TFS Buffer*	+	+0			-				
44	*Total Predicted Resource*	+	+0			-				

Attribute #	*Attribute Name*	*Run Node Executable Command*	*Configure Command*	*Operate Command*	*Halt Operations Command*	*Terminate Node Command*	*Configuration Test***	*Connection Test***	*Handle Fatal Operational Event*	*Handle Non-Fatal Operational Event*
	Buffer									
45	*Inelastic Load Prediction Buffer*	+	+0			-				
46	*Elastic Load Prediction Buffer*	+	+0			-				
47	*Predicted Inelastic and Elastic Load Buffer*	+	+0			-				
4	*Geographical Location of Node*	+	+0			-				
3	*Node Type*	+	+0			-				
8	*Mode*	+	+0	+0	+0	-			+0	+0
16	*Electrical Topology Location*	+	+0			-				
21	*Processing Time Delay*	+	+0			-				
22	*Time Out*	+	+0			-			(C)	(C)

* These Node attributes will be checked and should be configured (not empty) during a *Configuration Test.*

The Configuration and Connection Tests will additionally check the Asset attribute ***38* – *List of Local Information* and the statuses of connections.

"C" = condition checked; "(C)" = condition possibly checked; "++" = "should establish new attribute content"; "+" = "may establish new attribute content"; "- -" = "should remove existing attribute content; "-" = "may remove existing attribute content"; "**00** = "should modify existing attribute content"; "**0**" = "may modify existing attribute content"

6.1.7 Functions and Events of the Transactive Node State Model

Run Node Executable(Filename) Command

Command Parameters

- *Filename*—Filename that should be found in and run from a known file directory. If *Filename* cannot be found, fail in condition F**1**.

Command Logic

If *Filename* cannot be recognized or located, then reply

> "Command failed – (F**1**) File could not be found"

If this transactive node is already running an executable file, as can be determined by transactive node attribute ***7*** - *Node Status* being in a valid, defined state or other evidence that the executable is running, then reply

> "Command failed – (F**2**) Node executable is already running."

If the entity that made this command is not the local system manager and is not found to have been granted permission to make this command by attribute ***31*** - *Connection Partner's System Management Permissions*, then reply

> "Command failed – (F**3**) Lacking permission to make this command"

If after running Filename the attributes ***1*** – *Node ID*, ***5*** – *Node Version*, ***7*** – *Node Status*, and ***18*** - *Time* have not become configured, then do not run the node executable. Reply

> "Command failed – (F**4**) Critical transactive node attributes were not configured"

If the node executable fails to run for any other reason, reply

> "Command failed – (F**5**) Unknown reason"

Otherwise,

> The node executable runs to completion and its process remains active, including the management present time ***18*** – *Time* in UTC format.

Set attribute ***7*** *– Node Status* = "**2**" *(*state **2** *– Under Local Control).*

Populate attributes ***1*** *– Node ID*, ***5*** *– Node Version*, and ***18*** *– Time* with the contents supplied by *Filename*. These attributes may not be empty at the successful conclusion of this command.

Additionally, any other attribute may be populated at the time the node executable is run..

Reply, "Command succeeded – (S1)"

Configure() (Node Attributes) Command—a flexible command that is applicable to the transactive node as well as to the other connections that a transactive node manages. An important concept in the use of this command is that the connection's identifier should be stated before any of its attributes may be modified. Because this section is addressing only the transactive node state model, the only attributes that will be addressed in this section are transactive node attributes for this transactive node.

Command Parameters

ConfigureFile = (*Filename*) — If a file is named using this parameter, a command script will be read from *Filename* found in a known file directory. It is recommended that *Filename* should contain scripted parameters as would be used in line with the command.

Any combination of the following comma-separated, in-line command parameters may be used and in any order:

Node = (***1*** *- Node ID*) — (Optional) Should match the identity of this transactive node.

NodeAttribute **=** attribute #, attribute value **1**[[, attribute value **2**], …] — This parameter may be used to initialize or change the contents of any Node group attribute except attribute ***1*** *- Node ID*, ***5*** *– Node Version*, ***7*** *– Node Status*, or ***18*** *- Time.*

Command Logic

If the entity that made this command is not the local system manager and if the entity has not been explicitly given permission to make this system management command among the commands in its ***31*** ***–*** *Connection Partner's System Management Permissions*, then reply

"Command failed – (F**1**) Permissions do not include this command"

If attribute ***7*** – *Node Status* = "**5**" (state ***5*** – *Operational*), then reply

"Command failed – (F**2**) Configure command not allowed from Operational state."

If *Filename* cannot be found, reply

"Command failed – (F**3**) File cannot be found or opened"

If the *Node ID* does not match the presently configured *Node ID*, then reply

"Command failed – (F**4**) Incorrect node ID"

If the node attribute number does not match a known Node attribute number (e.g., is not a member of {**3**, **4**, **8**, **9**, **16**, **21**, **22**, **34**, **38**, **39**, **40**, **41**, **42**, **43**, **44**, **45**, **46**, **47**, **49**, **50** or **51**}), then reply

"Command failed – (F**5**) Command did not address known node attributes"

If the command cannot be completed for any other reason, reply

"Command failed – (F**6**) Unknown reason"

Otherwise,

Reply, "Command succeeded – (S**1**)"

Finalize any changes to transactive node attributes that were specified in the file or on the command line.

Run a *Configuration Test*.

Run a *Connection Test*.

Configuration Test()—this is neither a system management command nor an event, but it is a test of the present configuration that should be conducted automatically by a transactive node after a successful *Configure()* command. It is permissible that the test may be run more often, but the outcome should not be expected to change unless a successful *Configure()* command occurs.

Parameters—None.

Test Logic

If upon checking attribute **7** – *Node Status*, this transactive node is found to be in state "**5**" (**5** - *Operational*), then

Test passed - (S**1**) The *Operational* state is necessarily *Configured*.

No further tests are required. No state transition occurs. No attributes are changed.

If any of the attributes ***1*** – *Node ID*, **5** – *Node Version*, **7** – *Node Status*, **9** – *Update Frequency*, ***18*** – *Time*, **57** – *Interval Durations*, or **58** – *Numbers of Intervals* have not yet been configured and are therefore empty, then

Test failed – (F**1**) The transactive node is not configured.

Attribute **7** – *Node Status* = "**2**" (state **2** – *Under Local Control*).

If for any transactive neighbor connection listed in ***49*** – *List of Transactive Neighbors* a corresponding ***52*** – *Transactive Neighbor ID* has not been established, then

Test failed – (F**2**) Not all transactive neighbors have been *Listed*.

Attribute **7** – *Node Status* = "**2**" (state **2** – *Under Local Control*).

If for any system manager connection listed in ***50*** – *List of System Managers* a corresponding ***53*** – *System Manager ID* has not been established, then

Test failed – (F**3**) Not all system managers have been *Listed*.

Attribute **7** – *Node Status* = "**2**" (state **2** – *Under Local Control*).

If for any asset connection listed in ***51*** – *List of Assets* a corresponding **2** – *Asset ID* has not been established, then

Test failed – (F**4**) Not all assets have been *Listed.*

Attribute **7** – Node Status = "**2**" (state **2** – *Under Local Control*).

If for any *Listed* transactive neighbor (e.g., one for which a ***52*** – *Transactive Neighbor ID* has become established) its ***32*** – *Connection Status* is either undefined or "**1**" (connection state ***1*** – *Listed*), then

Test failed – (F**5**) Not all transactive neighbors have become configured.

Attribute **7** – *Node Status* = "**2**" (state **2** – *Under Local Control*).

If for any *Listed* system manager (e.g., one for which a ***53*** – *System Manager ID* has become established) its ***32*** – *Connection Status* is either undefined or "**1**" (connection state ***1*** – *Listed*), then

Test failed – (F**6**) Not all system managers have become configured.

Attribute **7** – *Node Status* = "**2**" (state **2** – *Under Local Control*).

If for any *Listed* asset (e.g., one for which a **2** – *Asset ID* has become established) its ***32*** – *Connection Status* is either undefined or "**1**" (connection state ***1*** – *Listed*), then

Test failed – (F**7**) Not all assets have become configured.

Attribute **7** – *Node Status* = "**2**" (state **2** – *Under Local Control).*

If for any asset connection that has local information connections listed in its ***38*** – *List of Local Information* a corresponding ***52*** – *Transactive Neighbor ID* has not been established, then

Test failed – (F**8**) Not all local information sources have been *Listed.*

Attribute **7** – *Node Status* = "**2**" (state **2** – *Under Local Control*).

If for any *Listed* local information connection (e.g., one for which a **48** – *Local Information ID* has become established) its ***32*** – *Connection Status* is either undefined or "**1**" (connection state ***1*** – *Listed*), then

Test failed – (F**9**) Not all local information connections have become configured.

Attribute **7** – *Node Status* = "**2**" (state **2** – *Under Local Control*).

If the *Configuration Test* fails to run to completion for any other reason, then

Test failed – (F**10**) Unknown reasons.

Attribute **7** – *Node Status* = "**2**" (state **2** – *Under Local Control*)

Otherwise,

Test passed – (S**2**).

If prior **7** – *Node Status* = "**2**" (state **2** – *Under Local Control*), then *Node Status* = "**3**" (state **3** – *Configured*).

Otherwise, *Node Status* should remain unchanged in the prior state.

Connection Test()—this is neither a system management command nor an event, but it is a test of the completeness of the connections that should be completed between this transactive node and its connections. A *Connection Test* should be conducted automatically by a transactive node after a successful *Configure()* command and after any connection changes its connection state. A transactive node should have passed a *Configuration Test* before a *Connection Test* may be passed.

Parameters—None.

Test Logic

If upon checking attribute ***7*** *– Node Status* this transactive node is found to be in state "**2**" (**2** *– Under Local Control*), then

Test failed - (F**1**) A transactive node should be *Configured* prior to a *Connection Test.*

No further tests are required.

If for any ***52*** *– Transactive Neighbor ID* its ***32*** *– Connection Status* is other than "**3**" (connection state ***3*** *– Connected*) or "**4**" (connection state ***4*** *– Lost Connection*), then

Test failed – (F**2**) Not all transactive neighbors are *Connected.*

Attribute ***7*** *– Node Status* = "**3**" (state ***3*** *– Configured*).

If for any ***53*** *– System Manager ID* its ***32*** *– Connection Status* is other than "**3**" (connection state ***3*** *– Connected*) or "**4**" (connection state ***4*** *– Lost Connection*), then

Test failed – (F**3**) Not all system managers are *Connected.*

Attribute ***7*** *– Node Status* = "**3**" (state ***3*** *– Configured*).

If for any ***2*** *– Asset ID* its ***32*** *– Connection Status* is other than "**3**" (connection state ***3*** *– Connected*) or "**4**" (connection state ***4*** *– Lost Connection*), then

Test failed – (F**4**) Not all assets are *Connected.*

Attribute ***7*** *– Node Status* = "**3**" (state ***3*** *– Configured*).

If for any ***26*** *– Local Information ID* its ***32*** *– Connection Status* is other than "**3**" (connection state ***3*** *– Connected*) or "**4**" (connection state ***4*** *– Lost Connection*), then

Test failed – (F**5**) Not all local information connections are *Connected.*

Attribute ***7*** *– Node Status* = "**3**" (state ***3*** *– Configured*).

If the Connection Test fails to run to completion for any other reason, then

Test failed – (F**6**) Unknown reason.

Attribute ***7*** – *Node Status* = "**3**" (state ***3*** – *Configured*).

Otherwise,

Test passed – (S**1**).

If prior ***7*** – *Node Status* = "**3**" (state ***3*** – *Configured*), then *Node Status* = "**4**" (state **4** – *Connected & Configured*).

Otherwise, *Node Status* should remain unchanged in its prior state.

Operate() Command

Command Parameters—None.

Command Logic

The entity making the command should be found to be this transactive node or one of its connections. If the entity making this system management command is not the local system manager and does not explicitly have this command listed among the commands in its ***31*** – *Connection Partner's System Management Permissions*, then reply

"Command failed – (F**1**) Permissions do not include this command."

If upon reviewing **7** – *Node Status* the transactive node is found to be in a state other than "**4**" (state ***4*** – *Connected & Configured*) or "**5**" (state ***5*** – *Operational*), then reply

"Command failed – (F**2**) This command is not allowed from current state."

If upon receiving this command this transactive node is not able to enter or remain in state ***5*** – *Operational* for any reason, then reply

"Command failed – (F**3**) Unknown reason"

Otherwise,

Reply, "Command succeeded – (S**1**)."

Set **7** – *Node Status* = "**5**" (state **5** – *Operational).*

Begin interacting with transactive neighbor connections and other connections that are managed at this transactive node according to the algorithms of the toolkit framework and functions.

Handle Fatal Operational Event / Handle Non-Fatal Operational Event

The following error categories have been identified:

- Application errors—an application error occurs within the transactive control toolkit and may be due to faulty software, logic or algorithms
- Security and signal validation errors:—security and signal validation errors are primarily associated with the incoming TIS and TFS signals
- Network errors—network errors are related to communications network connectivity between transactive nodes.

Each error in these categories can further be classified as transient ("Non-Fatal") or permanent ("Fatal").

A non-fatal error is an error where the system can recover from the error without significant degradation of system functionality and can therefore remain in the *Operational* state. For example, if a transactive node does not receive a TIS signal within the update interval (**5** minutes for the Demonstration), the TIS signal can be still be generated with minimal loss of functionality (refer to the toolkit framework for how this is accomplished). But if the TIS signal is not received for a number of hours, then the transactive node may consider this a fatal error and exit an *Operational* state. The function *Handle Non-Fatal Operational Event()* has been provided within this state model for the diagnostic recognition of and response to non-fatal errors that will occur while the transactive node is in an *Operational* state.

If a transient error happens often enough or lasts a long time it will turn into a fatal error. Fatal errors are, by definition, not recoverable and cause a transactive node to exit an *Operational* state. One of the two categories of fatal errors is due to a severe security, application, or network failure. A second category occurs when a non-fatal error is repeated "N" times in a row, or "K" times in an "M" minute interval depending on local policies. The function *Handle Fatal Operational Event()* has been provided within this state model for the diagnostic recognition of and response to fatal errors that may occur while the transactive node is in an *Operational* state.

The logic and details for these events remain to be worked out, but at this point the logic and details should be made to work within the state model that is being described here.

Halt Operations() Command

Command Parameters—None.

Command Logic

The entity making the command should be found to be this transactive node or one of its connections. If the entity making this system management command is not the local system manager and does not explicitly have this command listed among the commands in its ***31*** *– Connection Partner's System Management Permissions*, then reply

"Command failed – (F**1**) Permissions do not include this command."

If upon reviewing 7 *– Node Status* the transactive node is found to be in a state other than "**5**" (state ***5*** *– Operational*), then reply

"Command succeeded – (S1) Operations already halted."

Otherwise,

Reply, "Command succeeded – (S**2**)."

Set **7** *– Node Status* = "**4**" (state **4** *– Connected & Configured).*

Halt interacting with transactive neighbor connections and other connections that are managed at this transactive node according to the algorithms of the toolkit framework and functions.

Terminate Node() Command

Command Parameters—None.

Command Logic

If the entity that made this command is not the local system manager and is not found to have been granted permission to make this command by attribute ***31*** - *Connection Partner's System Management Permissions*, then reply

"Command failed – (F**1**) Lacking permission to make this command"

If upon checking ***7*** – *Node Status*, this transactive node is found to be in a state other than "**2**" (state ***2*** – *Under Local Control*) or "**3**" (state ***3*** – *Connected*), then reply

"Command failed – (F**2**) Command not accepted in present transactive node state"

If the node executable fails to run for any other reason, reply

"Command failed – (F**3**) Unknown reason"

Otherwise,

(Optional) Save a copy of the prior configuration. This configuration may be reloaded the next time a node executable is run to jump start the maturity of its configuration. This is the condition of the final state of this transactive node ***1*** - *New or Terminated.*

Stop the node executable process from running. Attributes may become undefined by this action.

Reply, "Command succeeded – (S**1**)".

6.1.8 Transactive Node State Transition Table

In the table below, the numbering convention used for these functions and events are concatenations of the prior and end states. Where multiple functions and events have identical prior and end states, letters have been appended. For example, "**54**b" is the number applied to the second of two transitions from state number **5** to state number **4**.

Table 9. State Transition Table for a Transactive Node

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
11	*Fail to Run Node Executable*	***1*** - *New / Terminated*	*Filename* parameter Source of command Attributes ***7*** - *Node Status* and ***31*** - *Connection Partner's System Management Permissions*		***1*** - *New / Terminated*	Reply: "Command failed – [(**F1**) File could not be found / (F3) Lacking permission to make this command / (**F4**) Critical transactive node attributes were not configured / (F**5**) Unknown reasons]" Command log entry	Failure - [(F**1**) File could not be found / (F3) Lacking permission to make this command / (**F4**) Critical transactive node attributes were not configured / (F**5**) Unknown reasons]	Command log entry
12	*Run Node Executable*	**1** - *New / Terminated* (starting state)	*Filename* parameter Source of command Attributes ***7*** - *Node Status* and ***31*** - *Connection Partner's System Management Permissio*	***1*** - *Node ID*, **5** - *Node Version*, ***7*** - *Node Status* = "**2**" (state **2** - *Under Local Control*), and ***18*** – *Time* should be configured. Any and	**2** - *Under Local Control*	Reply: "Command succeeded – (S**1**)" Action: Node executable runs Command log entry	Success – (S**1**). Node executable runs.	Command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			ns	all remaining attributes may be set.				
21	*Terminate Node*	**2** - *Under Local Control*	Source of command Attributes ***7** – Node Status* and ***31** – Connection Partner's System Management Permissions*	All attributes revert to an undefined state and are lost when the node executable is terminated.	***1** - New / Terminated* (final state)	Reply: "Command succeeded – (S**1**)" Action: Node executable terminated Command log entry	Success – (S**1**) Node executable successfully terminated	Command log entry
22a	*Configuration Test Failed*	**2** - *Under Local Control*	***7** – Node Status*, ***49** - List of Transactive Neighbors*, ***50** – List of System Managers*, ***51** – List of Assets*, ***38** – List of Local Information*, ***52** – Transactive Neighbor ID*, ***2** – Asset ID*, ***53** – System Manager ID*, ***48** – Local Information ID*, ***32** – Connectio*		**2** - *Under Local Control*	Test log entry	Failure – [(F**1**) The transactive node is not configured / (F**2**) Not all transactive neighbors have been Listed / (F**3**) Not all system managers have been listed / (F**4**) Not all assets have been listed / (F**5**) Not all transactive neighbors have become configured / (F**6**) Not all system managers have become configured / (F**7**) Not all	Test log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			n Status				assets have become configured / (F**8**) Not all local information connections have been Listed / (F**9**) Not all informations have become configured / (F**10**) Unknown reasons]	
22b	*Configure (Node Attributes)*	**2**. *Under Local Control*	Source of command; Command-line parameters; List of node attributes that may be configured; Attributes **7** – *Node Status* and **31** - *Connection Partner's System Management Permissions; referenced configuration file*	Node attributes in the following set may be set or modified (e.g., "configured"): {**3**, **4**, **8**, **9**, **16**, **21**, **22**, **34**, **38**, **39**, **40**, **41**, **42**, **43**, **44**, **45**, **46**, **47**, **49**, **50** or **51**}	**2**. *Under Local Control*	Reply: "Command succeeded – (S**1**)" Command log entry	Success – (S**1**)	Command log entry
22c	*Connection Test Failed*	**2** – *Under Local Control*	**7** – *Node Status*, **52** - *Transactive Neighbor ID*, **53** - *System Manager ID*, **2** –		**2** – *Under Local Control*	Test log entry	Test failed – (F**1**) A transactive node should be *Configured* prior to a *Connection Test*	Test log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			Asset ID, ***48*** *- Local Information ID*, ***32*** *- Connection Status*					
22d	*Fail to Configure* (Node Attributes)	**2** - *Under Local Control*	Source of command; Command-line parameters; List of node attributes that may be configured; *Attributes* ***7*** *– Node Status* and ***31*** *- Connection Partner's System Management Permissions;* Referenced *Filename.*		**2** - *Under Local Control*	Reply: "Command failed – [(**F1**) Permissions do not include this command / (F**3**) File cannot be found or opened / (F**4**) Incorrect node ID / (**F5**) Command did not address known node attributes / (F**6**) Unknown reason]" Command log entry	Failure - [(F**1**) Permissions do not include this command / (**F3**) File cannot be found or opened / (F**4**) Incorrect node ID / (**F5**) Command did not address known node attributes / **(F6)** Unknown reason]	Command log entry
22e	*Fail to Operate*	**2** *– Under Local Control*	Source of command; Attributes ***7*** *– Node Status* and ***31*** *- Connection Partner's System Management Permissions*		**2** *– Under Local Control*	<u>Reply</u>: "Command failed – [(F**1**) Permissions do not include this command / (**F2**) This command is not allowed from current	Failure – [(**F1**) Permissions do not include this command / (F**2**) This command is not allowed from current state]	Command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
						state]” Command log entry		
22f	*Fail to Halt Operations*	**2** – *Under Local Control*	Source of command, Attributes **7** – *Node Status* and **31** - *Connection Partner's System Management Permissions*		**2** – *Under Local Control*	Reply: “Command failed – (F1) Permissions do not include this command” Command log entry	Failure – (F**1**) Permissions do not include this command	Command log entry
22g	*Fail to Run Node Executable*	**2** – *Under Local Control*	*Filename* parameter; Source of command; Attributes **7** - *Node Status* and **31** - *Connection Partner's System Management Permissions*		**2** – *Under Local Control*	Reply: “Command failed – [(F**1**) File could not be found / (F**2**) Node executable Is already running]” Command log entry	Failure - [(F**1**) File could not be found / (F**2**) Node executable is already running]	Command log entry
22h	*Fail to Terminate Node*	**2** - *Under Local Control*	Source of command; Attributes **7** – *Node Status* and **31** – *Connection Partner's System Management Permissions*		**2** - *Under Local Control*	Reply: “Command failed – [(F**1**) Lacking permission to make this command / (F**3**) Unknown reason]” Command log entry	Failure - [(F**1**) Lacking permission to make this command / (F**3**) Unknown reason]	Command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
22i	*Halt Operations*	**2** – *Under Local Control*	Source of command; Attributes **7** – *Node Status* and **31** - *Connection Partner's System Management Permissions*		**2** – *Under Local Control*	Reply: “Command succeeded – (S1)” Command log entry	Success – (S**1**)	Command log entry
23	*Configuration Test Passed (condition (S2))*	**2** - *Under Local Control*	Attributes **7** – *Node Status*, **49** - *List of Transactive Neighbors*. **50** – *List of System Managers*, **51** – *List of Assets*, **38** – *List of Local Information*, **52** – *Transactive Neighbor ID*, **2** – *Asset ID*, **53** – *System Manager ID*, **48** – *Local Information ID*, **32** – *Connection Status*	**7** – *Node Status* = “**3**” (state **3** - *Configured)*	**3** - *Configured*	Test log entry	Pass condition (S**2**). See test logic. Transactive node configuration is complete and internally consistent.	Test log entry
31	*Terminate Node*	**3** - *Configured*	Source of command; Attributes **7** - *Node Status* and **31** - *Connection*	All attributes revert to an undefined state and are	**1** - *New / Terminated* (final state)	Reply: “Command succeeded – (S1)” Action: Node	Success – (S1) Node executable is terminated.	Command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			Partner's System Management Permissions	lost when the node executable is terminated.		executable stops Command log entry		
32	*Configuration Test Failed*	**3** - *Configured*	Attributes **7** – *Node Status*, **49** - *List of Transactive Neighbors*, **50** – *List of System Managers*, **51** – *List of Assets*, **38** – *List of Local Information*, **52** – *Transactive Neighbor ID*, **2** – *Asset ID*, **53** – *System Manager ID*, **48** – *Local Information ID*, **32** – *Connection Status*	**7** – *Node Status* = "**2**" (state **2** – *Under Local Control*)	**2** - *Under Local Control*	Test log entry	Failure – [(F**1**) The transactive node is not configured / (F**2**) Not all transactive neighbors have been Listed / (F**3**) Not all system managers have been Listed / (F**4**) Not all assets have been Listed / (F**5**) Not all transactive neighbors have become configured / (F**6**) Not all system managers have become configured / (F**7**) Not all assets have become configured / (F**8**) Not all local information connections have been Listed / (F**9**) Not all information connections have become configured / (F**10**) Unkno	Test log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
							wn reasons]	
33a	*Configuration Test Passed (condition (S2)*	*3 - Configured*	Attributes **7** – *Node Status*, **49** - *List of Transactive Neighbors*, **50** – *List of System Managers*, **51** – *List of Assets*, **38** – *List of Local Information*, **52** – *Transactive Neighbor ID*, **2** – *Asset ID*, **53** – *System Manager ID*, **48** – *Local Information ID*, **32** – *Connection Status*		*3 - Configured*	Test log entry	Pass condition (**S2**). See test logic. Transactive node configuration is complete and internally consistent.	Test log entry
33b	*Configure (Node Attributes)*	*3 - Configured*	Source of command; Command-line parameters; List of node attributes that may be configured; Attributes **7** – *Node Status* and **31** - *Connection Partner's System Managem*	Node attributes in the following set may be set or modified (e.g., "configured"): {**3**, **4**, **8**, **9**, **16**, **21**, **22**, **34**, **38**, **39**, **40**, **41**, **42**, **43**, **44**, **45**, **46**, **47**, **49**, **50** or	*3 - Configured*	Reply: "Command succeeded – (**S1**)" Command log entry	Success – (**S1**)	Command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			ent Permissions; Referenced configuration file *Filename*	**51**}				
33c	*Connection Test Failed*	**3** - *Configured*	Attributes **7** – *Node Status*, **52** - *Transactive Neighbor ID*, **53** - *System Manager ID*, **2** – *Asset ID*, **48** - *Local Information ID*, **32** - *Connection Status*		**3** - *Configured*	Test log entry	Test failed – [(F**2**) Not all transactive neighbors are Connected / (F3) Not all system managers are Connected / (F**4**) Not all assets are Connected / (F**5**) Not all local information connections are Connected / (F**6**) Unknown reason]	Test log entry
33d	*Fail to Configure* (Node Attributes)	**3** - *Configured*	Source of command; Command-line parameters; List of node attributes that may be configured; Attributes **7** – *Node Status* and **31** - *Connection Partner's System Management Permissions*;		**3** - *Configured*	Reply: "Command failed – [(F**1**) Permissions do not include this command / (F**3**) File cannot be found or opened / (F**4**) Incorrect node ID / (F**5**) Command did not address known node attributes / (F**6**) Unknown	Failure - [(F**1**) Permissions do not include this command / (F**3**) File cannot be found or opened / (F**4**) Incorrect node ID / (F**5**) Command did not address known node attributes / (F**6**) Unknown reason]	Command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			referenced configuration file			reason]" Command log entry		
33e	*Fail to Operate*	**3** - *Configured*	Source of command; Attributes **7** – *Node Status* and **31** - *Connection Partner's System Management Permissions*		**3** - *Configured*	Reply: "Command failed – [(F**1**) Permissions do not include this command / (F**2**) This command is not allowed from current state]" Command log entry	Failure – [(F**1**) Permissions do not include this command / (F**2**) This command is not allowed from current state]	Command log entry
33f	*Fail to Halt Operations*	**3** - *Configured*	Source of command, Attributes **7** – *Node Status* and **31** - *Connection Partner's System Management Permissions*		**3** - *Configured*	Reply: "Command failed – (F**1**) Permissions do not include this command" Command log entry	Failure – (F**1**) Permissions do not include this command	Command log entry
33g	*Fail to Run Node Executable*	**3** - *Configured*	*Filename parameter*; Source of command; Attributes **7** - *Node Status* and **31** - *Connection Partner's System*		**3** - *Configured*	Reply: "Command failed – [(F**1**)) File could not be found / (F**2**) Node executable is already running]" Command	Failure– [(F**1**)) File could not be found / (F**2**) Node executable is already running]	Command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			Management Permissions			log entry		
33h	*Fail to Terminate Node*	***3*** - *Configured*	Source of command; Attributes ***7*** – *Node Status* and ***31*** – *Connection Partner's System Management Permissions*		***3*** - *Configured*	Reply: "Command failed – [(F**1**) Lacking permission to make this command / (F**3**) Unknown reason]" Command log entry	Failure - [(F**1**) Lacking permission to make this command / (F**3**) Unknown reason]	Command log entry
33i	*Halt Operations*	***3*** - *Configured*	Source of command; Attributes ***7*** – *Node Status* and ***31*** - *Connection Partner's System Management Permissions*		***3*** - *Configured*	Reply: "Command succeeded – (S**1**)" Command log entry	Success – (S**1**)	Command log entry
34	*Connection Test Passed*	***3*** - *Configured*	Attributes ***7*** – *Node Status*, ***52*** - *Transactive Neighbor ID*, ***53*** - *System Manager ID*, ***2*** – *Asset ID*, ***48*** - *Local Information ID*, ***32*** - *Connection*	***7*** – *Node Status* = "**4**" (*state* ***4*** – *Connected & Configured*)	***4*** - *Connected & Configured*	Test log entry	Success – (S**1**). Set of connections is complete and connected	Test log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			Status					
42	*Configuration Test Failed*	**4** – *Connected & Configured*	Attributes **7** – *Node Status*, **49** - *List of Transactive Neighbors*, **50** – *List of System Managers*, **51** – *List of Assets*, **38** – *List of Local Information*, **52** – *Transactive Neighbor ID*, **2** – *Asset ID*, **53** – *System Manager ID*, **48** – *Local Information ID*, **32** – *Connection Status*	**7** – *Node Status* = "**2**" (*state* **2** – *Under Local Control*)	**2** - *Under Local Control*	Test log entry	Failure – [(F**1**) The transactive node is not configured / (F**2**) Not all transactive neighbors have been Listed / (F**3**) Not all system managers have been Listed / (F**4**) Not all assets have been Listed / (F**5**) Not all transactive neighbors have become configured / (F**6**) Not all system managers have become configured / (F**7**) Not all assets have become configured / (F**8**) Not all local information sources have been Listed / (F**9**) Not all information sourcess have become configured / (F**10**) Unknown reasons]	Test log entry
43	*Connection Test Failed*	**4** - *Connected & Configured*	Attributes **7** – *Node Status*, **52** - *Transactive*	**7** – *Node Status* = "**3**" (*state* **3** - *Config*	**3** - *Configured*	Test log entry	Test failed – [(F**2**) Not all transactive neighbors are connected /	Test log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			Neighbor ID, ***53*** *- System Manager ID,* ***2*** *– Asset ID,* ***48*** *- Local Information ID,* ***32*** *- Connection Status*	*ured)*			(F**3**) Not all system managers are connected / (F**4**) Not all assets are connected / (F**5**) Not all local information sources are Connected / (F**6**) Unknown reason]	
44a	*Configuration Test Passed (condition (S2)*	***4*** *– Connected & Configured*	Attributes ***7*** *– Node Status,* ***49*** *- List of Transactive Neighbors,* ***50*** *– List of System Managers,* ***51*** *– List of Assets,* ***38*** *– List of Local Information,* ***52*** *– Transactive Neighbor ID,* ***2*** *– Asset ID,* ***53*** *– System Manager ID,* ***48*** *– Local Information ID,* ***32*** *– Connection Status*		***4*** *– Connected & Configured*	Test log entry	Pass condition (S**2**). See test logic. Transactive node configuration is complete and internally consistent.	Test log entry
44b	*Configure (Node Attributes)*	***4*** *– Connected & Configured*	Source of command; Command-line parameters; List of	Node attributes in the following set may be set or	***4*** *– Connected & Configured*	Reply: "Command succeeded – (S**1**)" Command	Success – (S**1**) See command	Command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			node attributes that may be configured; Attributes **7** – *Node Status* and **31** - *Connection Partner's System Management Permissions*; Referenced configuration file *Filename*	modified (e.g., "configured"): {**3**, **4**, **8**, **9**, **16**, **21**, **22**, **34**, **38**, **39**, **40**, **41**, **42**, **43**, **44**, **45**, **46**, **47**, **49**, **50** or **51**}		log entry	logic	
44c	*Connection Test Passed*	**4** – *Connected & Configured*	Attributes **7** – *Node Status*, **52** - *Transactive Neighbor ID*, **53** - *System Manager ID*, **2** – *Asset ID*, **48** - *Local Information ID*, **32** - *Connection Status*		**4** - *Connected & Configured*	Test log entry	Success – (S1) All connections are complete and connected.	Test log entry
44d	*Fail to Configure* (Node Attributes)	**4** - *Connected & Configured*	Source of command; Command-line parameters; List of node attributes that may be configured		**4** - *Connected & Configured*	Reply: "Command failed – [(F1) Permissions do not include this command / (F3) File cannot be found or	Failure - [(F1) Permissions do not include this command / (F3) File cannot be found or opened / (F4) Incorrect node ID /	Command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			; Attributes **7** – *Node Status* and **31** - *Connection Partner's System Management Permissions; referenced configuration file*			opened / (F**4**) Incorrect node ID / (F**5**) Command did not address known node attributes / **(F6)** Unknown reason]” Command log entry	(F**5**) Command did not address known node attributes / **(F6)** Unknown reason]	
44e	*Fail to Operate*	**4** - *Connected & Configured*	Source of command; Attributes **7** – *Node Status* and **31** - *Connection Partner's System Management Permissions*		**4** - *Connected & Configured*	Reply: “Command failed – [(F**1**) Permissions do not include this command / **(F3)** Unknown reason]” Command log entry	Failure – [(F**1**) Permissions do not include this command / (F3) Unknown reason]	Command log entry
44f	*Fail to Halt Operations*	**4** - *Connected & Configured*	Source of command, Attributes **7** – *Node Status* and **31** - *Connection Partner's System Management Permissions*		**4** - *Connected & Configured*	Reply: “Command failed – (F1) Permissions do not include this command.” Command log entry	Failure – (F**1**) Permissions do not include this command	Command log entry
44g	*Fail to Run Node Executable*	**4** – *Connected & Configured*	*Filename parameter*; Source of command; Attributes **7** - *Node Status* and		**4** – *Connected & Configured*	Reply: “Command failed – **[(F1)** File could not be found / (F**2**) Node executable	Failure– [(F**1**) File could not be found / (F**2**) Node executable is already running]	Command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			31 *- Connection Partner's System Management Permissions*			is already running]" Command log entry		
44h	*Fail to Terminate Node*	***4*** *– Connected & Configured*	Source of command; Attributes ***7*** *– Node Status* and ***31*** *– Connection Partner's System Management Permissions*		***4*** *– Connected & Configured*	Reply: "Command failed – [(F**1**) Lacking permission to make this command / (F**2**) Command not accepted in present transactive node state]" Command log entry	Failure– [(F**1**) Lacking permission to make this command / (F**2**) Command not accepted in present transactive node state]"	Command log entry
44i	*Halt Operations*	***4*** *– Connected & Configured*	Source of command, Attributes ***7*** *– Node Status* and ***31*** *- Connection Partner's System Management Permissions*		***4*** *– Connected & Configured*	Reply: "Command succeeded – (S**1**)" Command log entry	Success – (S**1**)	Command log entry
45	*Operate*	**4** *- Connected & Configured*	Source of command; Attributes ***7*** *– Node Status* and ***31*** *- Connection Partner's*	***7*** *– Node Status* = "**5**" (state ***5*** *– Operational*)	**5** *- Operational*	Reply: "Command succeeded – (S**1**)" Action: Transactive node	Success – (S**1**)	Command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			System Management Permissions			begins interacting with transactive control and coordination system Command log entry		
53	*Connection Test Failed*	**5** - *Operational*	Attributes **7** – *Node Status*, **52** - *Transactive Neighbor ID*, **53** - *System Manager ID*, **2** – *Asset ID*, **48** - *Local Information ID*, **32** - *Connection Status*	**7** – *Node Status* = "**3**" (*state* **3** - *Configured*)	**3** - *Configured*	Test log entry	Test failed – [(F**2**) Not all transactive neighbors are Connected / (F**3**) Not all system managers are Connected / (F**4**) Not all assets are Connected / (F**5**) Not all local information sources are Connected / (F**6**) Unknown reason]	Test log entry
54a	*Handle Fatal Operational Event*	**5** - *Operational*	Diagnostic recognition of Fatal Operational Event Details TBD	**7** – *Node Status* = "**4**" (state **4** – *Connected & Configured*)	**4** - *Connected & Configured*	Notifications TBD Event log entry	Non-recoverable error during transactive node operations	Event log entry
54b	*Halt Operations*	**5** - *Operational*	Source of command; Attributes **7** – *Node Status* and **31** - *Connection Partner's System Management Permissio*	**7** – *Node Status* = "**4**" (state **4** – *Connected & Configured*)	**4** - *Connected & Configured*	<u>Reply</u>: "Command succeeded – (S**2**)" <u>Action</u>: The transactive node halts its interactions with the transactive	Success – (S**2**)	Command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			ns			control and coordination system Command log entry		
55a	*Configuration Test Passed (condition (S1)*	**5** - *Operational*	Attributes **7** – *Node Status*, **49** - *List of Transactive Neighbors*, **50** – *List of System Managers*, **51** – *List of Assets*, **38** – *List of Local Information*, **52** – *Transactive Neighbor ID*, **2** – *Asset ID*, **53** – *System Manager ID*, **48** – *Local Information ID*, **32** – *Connection Status*		**5** - *Operational*	Test log entry	Pass condition (S**1**). See test logic. Transactive node configuration is complete and internally consistent	Test log entry
55b	*Connection Test Passed*	**5** - *Operational*	Attributes **7** – *Node Status*, **52** - *Transactive Neighbor ID*, **53** - *System Manager ID*, **2** – *Asset ID*, **48** - *Local Informatio*		**5** - *Operational*	Test log entry	Success – (S1) All connections are complete and connected.	Test log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			n ID, ***32*** *- Connection Status*					
55c	*Fail to Configure* (Node Attributes)	**5** - *Operational*	Source of command; Command-line parameters; List of node attributes that may be configured; Attributes ***7*** *– Node Status* and ***31*** *- Connection Partner's System Management Permissions*; Referenced configuration file *Filename*		**5** - *Operational*	Reply: "Command failed – [(**F1**) Permissions do not include this command / (F**2**) Configure command not allowed from Operational state]" Command log entry	Failure - [(F**1**) Permissions do not include this command / (F**2**) Configure command not allowed from Operational state]	Command log entry
55d	*Fail to Halt Operations*	**5** - *Operational*	Source of command, Attributes ***7*** *– Node Status* and ***31*** *- Connection Partner's System Management Permissions*		**5** - *Operational*	Reply: "Command failed – (**F1**) Permissions do not include this command" Command log entry	Failure – (F**1**) Permissions do not include this command	Command log entry
55e	*Fail to Run Node Executable*	**5** - *Operational*	*Filename* parameter; Source of command;		**5** - *Operational*	Reply: "Command failed – [(**F1**) File could not	Failure– [(F1) File could not be found / (F**2**) Node executable is	Command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			Attributes **7** - *Node Status* and **31** - *Connection Partner's System Management Permissions*			be found / (**F2**) Node executable is already running]" Command log entry	already running]	
55f	*Fail to Terminate Node*	**5** - *Operational*	Source of command; Attributes **7** – *Node Status* and **31** – *Connection Partner's System Management Permissions*		**5** - *Operational*	Reply: "Command failed – [(**F1**) Lacking permission to make this command / (**F2**] Command not accepted in present transactive node state]" Command log entry	Failure– [(**F1**) Lacking permission to make this command / (**F2**] Command not accepted in present transactive node state]	Command log entry
55h	*Operate*	**5** - *Operational*	Source of command; Attributes **7** – *Node Status* and **31** - *Connection Partner's System Management Permissions*		**5** - *Operational*	Reply: "Command succeeded – (**S1**)" Command log entry	Success – (**S1**)	Command log entry

6.1.9 Connection Attributes

Connection attributes have been identified and are ascribable in common to the four types of connections. This set of attributes refers to a single connection between this transactive node and a transactive neighbor, system manager, asset system, or source of local information. The connection attributes are indispensible for keeping track of the state of any type of connection. It is never adequate to reference these attributes apart from a specific example of attribute ***27*** - *Connection ID*.

Connection attributes are important for navigating the connection state transition diagram **3000** in Figure **30.** The attribute **32** - *Connection Status* should be known and managed for each connection. Attribute **7** - *Node Status* has been shown to be a logical combination of multiple individual *Connection Statuses*.

Refer to Table **11** for the anticipated ways in which the connection attributes may be affected by the commands and events of the connection state model.

In the connection state model (see Figure **30**), a connection moves between its states by undergoing *Configuration Tests*, accepting *Connect* and *Disconnect* commands, and experiencing some events like *Loss of Connection*. Important connection attribute ***32*** – *Connection Status* keeps track of these state changes. For example, a local information connection transitions into state *Connection Status* "**2**" (connection state **2** – *Configured*) if connection attribute ***32*** - *Connection Status* and the local information attribute ***48*** – *Connection Status* have been configured. (The sets of attributes that should be configured before a connection may enter connection state **2** – Configured are indicated conveniently by asterisks in Figure **28.**)

Table 10. Dictionary of Connection Attributes that should be applied to each Connection

No.	Attribute Name	Description	Role	Type	Format	Range of values
32	*Connection*	Indicates the state of the connection	Affected by *Connect()*	Integer	Example: "**2**"	***1*** – *Listed,* ***2*** - *Configured,* ***3*** - *Connected,*

No.	Attribute Name	Description	Role	Type	Format	Range of values
	*Status**	between this transactive node and a connection.	command. A transactive node conducts a Connection Test based on the Connection Statuses of its connections.			***4*** - *Lost Connection*
29	*Connection Partner Type**	An indicator of type of connection partner from a list of allowed partner types to include at least transactive neighbors, owner.	May be used to indicate applicable interactions and permissions. For example, transactive neighbors expect to receive and supply transactive signals. System managers should be granted some system management permissions.	Character string	Example: "SM"	"RL" – Responsive Load "OL" – Other Local Condition Input "OS" – Owner or Subsystem "RR" – Responsive Rsource "SM" – System Manager "TN" – Transactive Neighbor
17	*Connection Details*	Optional additional details about the connection method stated in attribute ***33*** - *Connection Method*	Each connection method used in attribute ***33*** *- Connection Method* should prescribe a set of details that should	List of alphanumeric strings	Detail**1**, detail**2**, ...	A list of necessary details should be created for each connection method of attribute ***33***. For example,

No.	Attribute Name	Description	Role	Type	Format	Range of values
		for a connection.	be provided by this attribute.			Internet(IP address of this transactive node, IP address of connection partner, encryption level, ...)
28	*Connection's Geographical Location*	The locations of connection partners should be provided to identify map locations to which this transactive node has established connections. This attribute is optional for each connection.	Support future GIS system representations.	Most likely a pair of real numbers representing angular latitude and longitude.	(latitude, longitude) As for attribute **#4**. *Geographical Location*, angular latitude and longitude are the default units. Standard GIS representation formats should be adapted if such standards can be identified.	**0 – 360** degrees; **0 – 2***pi radians
30	*Entities Permitted to Modify this Connection*	For each specified connection, a list of those entities that are permitted to initiate, modify, or disconnect the connection and its attributes. This list may narrow the permissions granted to a connection partner by attribute ***31** - Connecti*	Eventually, the transactive nodes will operate with considerable autonomy and should clearly specify which, if any connection partners may modify connections. The Demonstration has many instances where a utility owner should be granted	List of alphanumeric identifiers, one for each entity that will be granted this permission for this connection.	Use guidance provided with ***1** - Node ID* and ***28** - Connection ID*. Use formats found in transactive control and coordination system topology maps.	If null, only the local transactive node system manager may modify the specified connection.

No.	Attribute Name	Description	Role	Type	Format	Range of values
		on Partner Permissions.	permissions to modify a transactive node's connections.			
31	*Connection Partner's System Management Permissions*	The general permissions granted to connection partners to issue system management commands at this transactive node, plus the transactive node attributes that may be modified by the connection partner during configuration. These permissions may be restricted further by attribute *30 - Entities Permitted to Modify this Connection*. .	This attribute allows system management responsibilities to be assigned to one or more of the connection partners at this transactive node. Assigned among Connection Table attributes. See Connection Table.	List of system management commands that will be accepted from a connection partner at this transactive node, plus list of transactive node attributes that may be modified by this connection partner.	List of allowed system management commands {command**1**, command**2**, …}. If the list includes command *Configure()*, the list of modifiable attributes should be listed as parameters of this command by number.	Entries selected from {*Configure*([All, **1**, **2**, …]), *Connect*, *Disconnect*, *Operate*, *Run Node Executable*, *Stop*, *Terminate Node*} If null, then only this transactive node's system manager may issue system management commands.
33	*Connection Method*	Optional indication of the media and protocol used in a connection.	Specify the method of connection. Each such method may then have specific details to be listed in attribute ***34** - Connecti*	Single character string	Example: "Internet"	Ethernet, Internet, Wireless Zigbee®, Wireless other, Power Line Carrier

No.	Attribute Name	Description	Role	Type	Format	Range of values
			on Details.			
54	*Connection Timeout Period*	The period of time that a given connection will remain in its *Lost Connection* state before it will terminate the connection, which could threaten the *Operational* status of this transactive node.	This is the amount of time that should elapse before a connection in its *Lost Connection* state will automatically transition back into its *Configured* state. This duration may be quite long if this transactive node and its algorithms have been designed tolerant of poor connectivity. This timeout period is to be individually configured for each connection.	Character string representation of a single time duration	Recommend "dd:hh:mm". Should emulate UTC standard format that is used frequently in state model.	The Demonstration should use values longer than **5** minutes "**00**:**00**:**05**" or shorter than **4** days "**04**:**00**:**00**". Default value: **1** hour: "**00**:**01**:**00**".
55	*Loss of Connection Event Buffer*	A list of times at which *Loss of Connection Events* have occurred for a given connection.	By keeping track of when *Loss of Connection Events* occur, a transactive node can take exceptional actions based on the frequency with which the events have occurred.	List of UTC times.	See UTC standard	Allow for cyclic buffer of **64** values. Need not be initialized.

No.	Attribute Name	Description	Role	Type	Format	Range of values
56	*Allowed Frequency of Loss of Connection Events*	The frequency with which *Loss of Connection Events* will be tolerated before the connection will be severed. There is a criterion for events per hour and another for events per day.	Criteria placed on the members of **55** – *Loss of Connection Event Buffer*. The connection should be severed if these frequencies are exceeded, which would indicate a problem with the connection.	Two integers.	Example: (**5**, **24**), meaning **5** times in an hour, or **24** times in a day	Default (**6**, **48**), meaning six times in an hour, or **48** times in during a day. Integers should be less than the buffer length of **55** - *Loss of Connection Event Buffer*.

Table 11. The Ways Connection Attributes May be Affected by Connection State Model Commands and Events

Attribute #	***Attribute Name***	***Configure Command***	***Configuration Test*****	***Connect Command***	***Disconnect Command***	***Loss of Connection Event***	***Re-Establish Connection Event***	***Terminate Connection Event***
32	*Connection Status**	**+0**	**C+0**	C0	C0	C00	C00	C00

Attribute #	***Attribute Name***	**Configure Command**	***Configuration Test*****	***Connect Command***	***Disconnect Command***	***Loss of Connection Event***	***Re-Establish Connection Event***	***Terminate Connection Event***
29	*Connection Partner Type**	**+0**		C		(C)		
30	*Entities Permitted to Modify this Connection*	**C+0**		C				
31	*Connection Partner's System Management Permissions*	**C+0**		C	C			
17	*Connection Details*	**+0**		(C)		(C)	(C)	(C)
28	*Connection's Geographical Location*	**+0**						
33	*Connection Method*	**+0**		(C)	(C)	(C)	(C)	(C)
54	*Connection Timeout Period*	**+0**				C	C	C
55	*Loss of Connection Event Buffer*	**+0**				C+0		
56	*Allowed Frequency of Loss of Connection Events*	**+0**				C		

*The Connection Status should be configured before a connection can enter its **2 –** *Configured* state.

**The connection *Configuration Test* additionally should check one or more attributes of the connection partner type.

6.1.10 Transactive Neighbor Connection Attributes

In certain embodiments, transactive node define at least one connection to a transactive neighbor. The connection may be observed and maintained using the union of connection attributes and transactive node attributes (see Figure **28**).

At least for some of the connections that are being made to transactive neighbors, it may be desired that experimenters and testing entities have the means to redirect the inputs received from the transactive neighbors so that these inputs may be received instead from selected alternative sources of such information. It is likewise important that one may redirect the output from these connection partners to one or more alternative locations. For the special type of connection partners called transactive neighbors, the means to redirect inputs and outputs has been accomplished with attributes **10** – **13**, which attributes define the sources and targets of transactive signals. The sources and targets are not necessarily the transactive neighbor itself. Using these attributes, simulations and "what-if" scenarios may be designed and tested in the production or test system environments. (So far, attributes **#10** – **13** only apply to transactive neighbors and their connections. It is conceivable that the attributes could be generalized and renamed to apply to any connection type, not only transactive neighbors.)

Table 12. Dictionary of Transactive Neighbor Attributes

No.	Attribute Name	Description	Role	Type	Format	Range of values
52	*Transactive Neighbor ID**	The identifier to be used for one transactive neighbor with which this transactive node will exchange electrical energy and therefore will exchange transactive signals.	This asset should be repeated for each member of ***49*** - *List of Transactive Neighbors* to instantiate the transactive neighbors that this transactive node expects to interact with. This transactive neighbor enters its *Listed* state after this attribute has been configured.	Single character string	Example #1: "UT06", which is the Demonstration's identifier for the Avista utility.	See system topology. Naming practice should be the same here and for attribute ***49*** - *List of Transactive Neighbors*.
10	*Receive TIS Source**	The *Connection ID* of a source from which a transactive neighbor's	This attribute permits alternative TIS examples to be received	Single, short alphanumeric identifier for each transactive neighbor.	Use guidance provided with ***1*** - *Node ID* and ***28*** - *Connection ID*. Use formats found in transactive	Source should be a known source within present transactive control and coordination system.

No.	Attribute Name	Description	Role	Type	Format	Range of values
		TIS should be received. The source is not necessarily the transactive neighbor itself.	at this transactive node from alternative sources to facilitate testing and simulation.		control and coordination system topology maps.	
11	*Receive TFS Source**	The *Connection ID* of a source from which a transactive neighbor's TFS should be received. The source is not the transactive neighbor itself.	This attribute permits alternative TFS examples to be received at this transactive node to facilitate testing and simulation.	Single, short alphanumeric identifier for each transactive neighbor	Use guidance provided with *1 - Node ID* and *28 - Connection ID*. Use formats found in transactive control and coordination system topology maps.	Source should be a known source within present transactive control and coordination system.
12	*Send TIS Targets**	The *Connection ID* of at least one target location to which this transactive node's TIS should be sent. The target location is not necessarily that of the transactive neighbor itself.	This attribute permits this transactive node's TIS to be sent to one or more places to facilitate testing and simulation.	List of one or many single short alphanumeric identifiers for each transactive neighbor	Use guidance provided with *1 - Node ID* and *28 - Connection ID*. Use formats found in transactive control and coordination system topology maps.	Target should be known location within present transactive control and coordination system.
13	*Send TFS Targets**	The *Connection ID* of at least one target location to which this transactive node's calculated TFS with this transactive	This attribute permits this transactive node's TFS for this transactive neighbor to be sent to one or more places to facilitate testing and	List of one or many single short alphanumeric identifiers for each transactive neighbor	Use guidance provided with *1 - Node ID* and *28 - Connection ID*. Use formats found in transactive control and coordination system topology maps.	Target should be known location within present transactive control and coordination system.

No.	Attribute Name	Description	Role	Type	Format	Range of values
		neighbor should be sent. The target location is not necessarily that of the transactive neighbor itself.	simulation.			
23	*Received TIS Buffer*	Contains at least the most recent TIS messages received from each transactive neighbor.	Each transactive neighbor's TIS is used within the toolkit framework algorithms. To be stored to the *Input Transactive Signal Buffer* of the toolkit framework.	List of TIS	According to transactive signal format as defined by approved XML schema for the TIS.	See range attributes of TIS
24	*Received TFS Buffer*	Contains at least the most recent TFS messages received from each transactive neighbor.	Each transactive neighbor's TFS is used within the toolkit framework algorithms. To be stored to the *Input Transactive Signal Buffer* of the toolkit framework.	List of TFS	According to transactive signal format as defined by approved XML schema for the TFS.	See range attributes of the TFS
59	*TIS Sent Flag*	Flag that is set if a *TIS* has been transmitted to this transactive neighbor connection by this transactive node during the current update interval.	This flag may be used in conjunction with the watchdog timer. The actions taken upon a watchdog timer event may desirably have the	Boolean condition flag: **0** – cleared **1** - set	Boolean logic.	**0** – default value – cleared - no TIS has been transmitted to this transactive neighbor yet during the current update interval. **1** – set – a TIS has been transmitted to this transactive neighbor during

No.	Attribute Name	Description	Role	Type	Format	Range of values
		The flag is cleared at the beginning of each update interval.	transactive node keep track of to which transactive neighbor transactive signals have been transmitted and not.			the current update interval.
60	*TFS Sent Flag*	Flag that is set if a *TFS* has been transmitted to this transactive neighbor by this transactive node during the current update interval. The flag is cleared at the beginning of each update interval.	This flag may be used in conjunction with the watchdog timer. The actions taken upon a watchdog timer event may desirably have the transactive node keep track of to which transactive neighbor transactive signals have been transmitted and not.	Boolean condition flag: set / cleared.	Boolean logic.	0 – default value – cleared - no TFS has been transmitted to this transactive neighbor yet during the current update interval. 1 – set – a TFS has been transmitted to this transactive neighbor during the current update interval.
*This attributes should be configured to pass a connection Configuration Test.						

Table 13. The Ways Transactive Neighbor Attributes May be Affected by Connection State Model Commands and Events

Attribute #	***Attribute Name***	***Configure Command***	***Configuration Test*****	***Connect Command***	***Disconnect Command***	***Loss of Connection Event***	***Re-Establish Connection Event***	***Terminate Connection Event***
52	*Transactive Neighbor*	(C)+0	C	(C)	(C)	(C)	(C)	(C)

Attribute #	Attribute Name	Configure Command	Configuration Test**	Connect Command	Disconnect Command	Loss of Connection Event	Re-Establish Connection Event	Terminate Connection Event
	*ID**							
10	*Receive TIS Source**	+0	C	(C)	(C)	(C)	(C)	(C)
11	*Receive TFS Source**	+0	C	(C)	C)	(C)	(C)	(C)
12	*Sent TIS Targets**	+0	C	(C)	(C)	(C)	(C)	(C)
13	*Send TFS Targets**	+0	C	(C)	(C)	(C)	(C)	(C)
23	*Received TIS Buffer*	+0						
24	*Received TFS Buffer*	+0						

*These attributes should be configured before a transactive neighbor connection can enter its **2** – *Configured* state.

The connection *Configuration Test* additionally should check that ***32* – *Connection Status* has been configured.

6.1.11 System Manager Connection Attributes

In certain embodiments, a single attribute can define a connection to a system manager.

Table 14. Ways in which System Manager Connection Attributes may be affected by Connection Commands and Events

Attribute #	Attribute Name	Configure Command	Configuration Test**	Connect Command	Disconnect Command	Loss of Connection Event	Re-Establish Connection Event	Terminate Connection Event
52	*System Manager*	(C)+0	C	(C)	(C)	(C)	(C)	(C)

Attribute #	*Attribute Name*	*Configure Command*	*Configuration Test***	*Connect Command*	*Disconnect Command*	*Loss of Connection Event*	*Re-Establish Connection Event*	*Terminate Connection Event*
	*ID**							

*The Connection Status should be configured before a connection can enter its **2** – *Configured* state.

**The connection *Configuration Test* additionally should check one or more attributes of the connection partner type.

Note that in certain implementations, transactive nodes establish and maintain a connection to the global system manager. Therefore, attribute **52** System Manager ID includes the ID of this global system manager for the transactive nodes.

Table 15. Dictionary of System Manager Connection Attributes

No.	Attribute Name	Description	Role	Type	Format	Range of values
52	*System Manager ID**	The identifier for one system manager. This entity will be granted permissions by this transactive node to make some system management commands.	This attribute instantiates a system manager from those that appear in ***50*** – *List of System Managers*. A system manager for which this attribute has been configured will enter its *Listed* state. A system manager is not a transactive neighbor, but a transactive neighbor may be granted permissions to act as a system manager. This transactive node may instantiate multiple connections to system managers. The Demonstration, for example, will have some central system management ("EI**01**"), but this transactive node may also grant system administration rights to the utility that "owns" this transactive node.	Single character strings	Example #**1**: "EI**01**" to represent the system manager, from which system management command will likely be received.	See system topology. Naming practice should be the same here and for attribute ***50*** – *List of System Managers*.

6.1.12 Asset Connection Attributes

This group of Asset attributes are meaningful only in respect to a given connection to an asset, which can be an energy resource, an incentive, or a load. Each resource or incentive has a corresponding toolkit resource and incentive function that defines how its behavior and effects may be modeled or predicted for the formulation of the transactive signals according to the toolkit framework. Each load similarly should have a corresponding toolkit load function that describes its effect on the formulation of the TFS. Often these "assets" will, in fact, be rather complex systems of assets.

An asset connection may list a set of local information connections that should be established via its ***38*** *– List of Local Information.* Each member of this list creates an expectation that a local information connection will become established.

An asset connection should have its **2** – *Asset ID,* **6** – *Toolkit Function*, and **6** – *Asset Type* configured before it is able to enter into the connection state **2** – *Configured*

Table 16. Dictionary of Asset Connection Attributes

No.	Attribute Name	Description	Role	Type	Format	Range of values
2	*Asset ID**	This attribute identifies the resources, incentives, and loads associated with this transactive control node.	Each resource, incentive, or load should be identified along with its toolkit function, status, predicted / scheduled engagement, etc.	Character string	Recommend format "XX-#" for each asset, where "XX" is a 2-letter acronym for the owner of this transactive node, and "#' is an integer number that ensures that the identifier is unique.	The Demonstration has already specified identifiers for responsive asset systems (most of which are loads) according to this convention. Loads = [**0-999**] Resources = [**1000-1999**] Incentives = [**2000-2999**]

No.	Attribute Name	Description	Role	Type	Format	Range of values
6	*Toolkit Function ID**	An identification of the specific toolkit function and version—the functional algorithms used at this transactive node to process the TIS, TFS, local information, and to control associated assets.	States the specific toolkit function and version that is being applied at this transactive node for each resource, incentive, or load. A toolkit function should be named for each resource, incentive, and load for which predictive behavior is being modeled by a toolkit function.	List of Alphanumeric modules or filenames and the present version of these modules.	{filename1, #.##; filename2, #.##, ...}	Valid filenames are to be used. "#.##" is major and minor version numbers using digits 0-9.
25	*Asset Output Targets**	Enables a control output to a resource or load to become redirected to one or more target locations. The target locations do not necessarily include the resource or load itself.	This feature may facilitate using the installed transactive control and coordination system for simulation of asset responses under alternative scenarios and during testing. If targets do not include the asset system, the asset should not respond. Should be configured for a successful connection *Configuration*	List of alphanumeric identifiers	See *2 – Asset ID*.	Should refer to valid resource or load entity ID from the *2 - Asset IDs* that are being used.

No.	Attribute Name	Description	Role	Type	Format	Range of values
			Test.			
36	*Asset Type*	Declaration of asset type at least from among "Resource," "Incentive," and "Load."	May be useful for categorization of asset connections. Range of values may be expanded. See the toolkit framework to understand the roles of toolkit functions.	Single alphanumeric string	Example: "Resource"	"Resource"—describes a generator resource at this transactive node. "Incentive"—describes an incentive that is not a resource. "Load"—describes an elastic or inelastic load at this transactive node.
38	*List of Local Information Connections*	A list of the sources of local information that will be called upon to help predict the behavior of this asset.	An asset's predicted behavior is modeled by a toolkit function, which in turn may call upon sources of local information. A connection listed in this attribute creates an expectation that this transactive node will establish and manage the connection.	List of character strings	See ***48** – Local Information ID.*	Should correspond to valid ***48** – Local Information ID.*

To support future simulations and testing, the connection state model includes an ability to redirect the output of these asset connections. Some of the assets will be responsive to the transactive control and coordination system and an output "control" signal is sent to these asset systems by this transactive node. Attribute ***25*** - *Asset Output Targets* allows the targets of these "control" signals to be sent to the asset system, to another entity, or to both the asset system the other entity.

List the local information inputs that are anticipated by an asset system and the toolkit function that predicts its behaviors. These streams of input information that are at time referred to as "other local conditions" should additionally become listed as attributes ***48*** – *Local Information ID* so that the continuity of the data stream may be monitored and so the input can become redirected, thus allowing alternative scenarios to be simulated with alternative input information.

Table **17** lists the asset attributes and indicates how these attributes may be affected by the system management commands and events that are part of the connection state model.

Table 17. The Ways Asset Attributes May be Affected by Connection State Model Commands and Events

Attribute #	***Attribute Name***	***Configure Command***	***Configuration Test*****	***Connect Command***	***Disconnect Command***	***Loss of Connection Event***	***Re-Establish Connection Event***	***Terminate Connection Event***
2	*Asset ID**	C+**0**	C	(C)	(C)	(C)	(C)	(C)
6	*Toolkit Function**	+**0**	C					
25	*Asset Output Targets**	(C)+**0**	C	(C)	(C)	(C)	(C)	(C)
36	*Asset Type*	+**0**	(C)					
38	*List of Local Information Connections*	(C)+**0**	C	(C)	(C)	(C)	(C)	(C)

*These attributes should be configured before an asset connection can enter its **2** – *Configured* state.

The connection *Configuration Test* additionally should check that ***32* – *Connection Status* has been configured.

The Assets in the Asset Table of Table **16** are closely aligned with several of the interim data storage areas ("buffers") that have been defined in the toolkit framework and with appear also in the state mode. For an asset connection there should be corresponding entries in one or more of the buffer (storage) areas that have been defined in the toolkit framework:

- Resource entries necessitate updating one record in attribute **34** - *Resource Schedule and Cost Buffer* during each iteration at the update frequency. (An exception may occur because an option has been provided for resource schedules to be entered without corresponding toolkit functions. This might be selected for some resources that are dispatched entirely unaffected by transactive control.) For a resource, this entry will state at least an energy parameter and average power produced by the corresponding resource for each interval start time.

- Incentive entries, like resources, also necessitate updating one record in attribute **34** - *Resource Schedule and Cost Buffer* during each iteration at the update frequency. That entry will include entries from among a paired set of capacity factor and capacity, an infrastructure parameter, and another costs parameter.

- Load entries necessitate one record be made in each attribute **45** - *Inelastic Load Prediction Buffer* and **46** - *Elastic Load Prediction Buffer* each iteration. The entries in those buffer (storage) locations predict load and, for responsive assets, the predicted level of engagement of responsive asset systems.

6.1.13 Local Information Connection Attributes

Table 18. Dictionary of Local Information Connection Attributes

No.	Attribute Name	Description	Role	Type	Format	Range of values
48	*Local Information ID**	Unique identifier to keep track of the local information that are used by this transactive node. "Local	This ID should be listed in a record of the Connections Table. Once clearly identified, this input may then be supplied by	Single character string.	Recommend "XX-OLC-**3###**", where XX is an acronym for the node owner, "**3###**" is a number from **3000** to **3999**. Example: "AV-	Should match formats and entries in ***38*** *– List of Local Information Connections*

No.	Attribute Name	Description	Role	Type	Format	Range of values
		Information" has been referred to as "Other Local Conditions" in the toolkit framework and in other sections.	alternative sources via the attribute 26 – *Local Information Source.*		OLC-3001"	
26	Local Information Source*'	One source of *Local Information* will normally be the actual source of the data. This attribute allows that the input data may be received from alternative sources.	Enables an alternative source of other local conditions to be used to facilitate testing and simulation.	Character string.	Example **1**: "AV**3015**" Example **2**: "EI**01**" Example **3**: "OLCFile**01**.exe"	Alternative **1**: ID of other local condition provider from Other Local Condition Table. Alternative **2**: Valid ID from among Connection Table records Alternative **3**: Valid filename in known directory.

A transactive node may possess many assets, and each asset may invoke multiple input information streams. Therefore, the local information connections should be carefully defined in the connection state model, and two attributes have been grouped as local information connection attributes.

A local information connection is an input that is invoked by and used by a toolkit function. Experimenters and testing personnel may wish to intentionally insert other alternative input information into the toolkit functions via this local information to simulate alternative scenarios that would be unlikely to occur under normal operations. Attribute ***48*** has been provided for this purpose, with which the source of the local information may be received from either the normal information provider or from an alternative source like an input file.

Table **19** lists which of the state model's commands and events are expected to modify the two Other Local Condition Attributes.

Table 19. Ways in Which Local Information Connection Attributes May be Affected by Commands and Events in this State Model

Attribute #	***Attribute Name***	***Configure Command***	***Configuration Test*****	***Connect Command***	***Disconnect Command***	***Loss of Connection Event***	***Re-Establish Connection Event***	***Terminate Connection Event***
48	*Local Information ID**	C+**0**	C	(C)	(C)	(C)	(C)	(C)
26	*Local Information Source**	+**0**	C	(C)	(C)	(C)	(C)	(C)

*These attributes should be configured before a local information connection can enter its ***2*** – *Configured* state.

The connection *Configuration Test* additionally should check that ***32* – *Connection Status* has been configured.

6.1.14 Functions and Events of the Connection State Model

Configure() (Connection Attributes) Command—the same flexible command that was applied to the transactive node may also be used for configuring the connections that a transactive node manages. Only the new parameters that should be used for connections will be presented; most parameters that were used for the transactive node state model will not be repeated. This command is used with connection attributes by first referring to the respective connection identifier (e.g., contents of attributes **52**, **2**, **53**, or **48**) and setting or modifying that connection's remaining attributes.

Command Parameters

ConfigureFile = (*Filename*) — If a file is named using this parameter, a command script will be read from *Filename* found in a known file directory. It is recommended that the *Filename* should contain scripted parameters as would be used an in-line command.

Any combination of the following comma-separated, in-line command parameters may be used and in any order:

***TransactiveNeighbor* = (*52* –** *Transactive Neighbor ID*) — If the *Transactive Neighbor ID* does not match an existing one, configure a new *Transactive Neighbor ID.*The commands that follow this command in the sequence of command parameters are assumed to refer to this transactive neighbor connection. This command parameter may be used again to reference another transactive neighbor connection.

- ***TransactiveNeighborDelete* —** Remove the record for the most recently referenced *Transactive Neighbor ID.*
- ***TransactiveNeighborAttribute*** = attribute #, attribute value **1**[[, attribute value **2**], ...] — This parameter may be used to initialize or change the contents of any transactive neighbor connection attribute except attributes ***52*** – *Transactive Neighbor ID* and ***32*** – *Connection Status..*

SystemManager* = (*53 – *System Manager ID*) — If the *System Manager ID* does not match an existing one, configure a new *System Manager ID*. The commands that follow this command in the sequence of command parameters are assumed to refer to this system manager connection. This command parameter may be used again to reference another system manager connection.

- ***SystemManagerDelete* —** Remove the record for the most recently referenced *System Manager ID*.
- ***SystemManagerAttribute*** = attribute #, attribute value **1**[[, attribute value **2**], ...] — This parameter may be used to initialize or change the contents of any system manager connection attribute except attributes ***53*** – *System Manager ID* and ***32*** – *Connection Status.*

Asset = (***2*** – *Asset ID*) — If the *Asset ID* does not match an existing one, configure a new *Asset ID*. the commands that follow this command in the sequence of command parameters are assumed to refer to this asset connection. This command parameter may be used again to reference another asset connection.

- ***AssetDelete*** — Remove the record for the most recently referenced *Asset ID*.
- ***AssetAttribute*** = attribute #, attribute value **1**[[, attribute value **2**], ...] — This parameter may be used to initialize or change the contents of any asset connection attribute except attributes ***2*** – *Asset ID* and ***32*** – *Connection Status*.

LocalInformation= (***48*** – *Local Information ID*) — If the *Local Information ID* does not match an existing one, configure a new *Local Information ID*. The commands that follow this command in the sequence of command parameters are assumed to refer to this local information connection. This command parameter may be used again to reference another local information connection.

- ***LocalInformationDelete*** — Remove the record for the most recently referenced *Local Information ID*.
- ***LocalInformationAttribute*** = attribute #, attribute value **1**[[, attribute value **2**], ...] — This parameter may be used to initialize or change the contents of any local information connection attribute except attributes ***48*** – *Local Information ID* and ***32*** – *Connection Status*.

Command Logic

If the entity that made this command is not the local system manager and if the entity has not been explicitly given permission to make this system management command among the commands in its ***31*** – *Connection Partner's System Management Permissions*, then reply

"Command failed – (F1) Permissions do not include this command"

From transactive node part of state model, which addressed the *Configure* function, if attribute ***7*** *– Node Status* = "**5**" (state ***5*** *– Operational*), then reply

> "Command failed – (F**2**) Configure command not allowed from Operational state."

If ***32*** *– Connection Status* is "**3**" (connection state ***3*** *– Connected*) or "**4**" (connection state ***4*** *– Lost Connection*), from which configuration of a connection is not be permitted, then reply

> "Command failed – (F**12**) Configure command not allowed from connected connection states."

If *Filename* cannot be found, reply

> "Command failed – (F**3**) File cannot be found or opened"

Failure conditions F**4** (Incorrect node ID) and F**5** (Command did not address known node attributes) do not apply during configuration of connections but should be reserved nonetheless.

If the entity making this system management command attempts to change a given connection's attributes, but the entity is not listed among this connection's ***30*** *– Entities Permitted to Modify this Connection* (applies to any of the types of connections), then reply

> "Command failed – (F**7**) Entity making command does not have permission to configure this connection."

If the transactive neighbor connection attribute number does not match a known transactive neighbor connection attribute number (e.g., is not a member of **{10**, **11**, **12**, **13**, **17**, **23**, **24**, **28**, **29**, **30**, **31**, **33}**), or if no ***52*** - *Transactive Neighbor ID* has been stated as a parameter before this command attempts to configure its attributes, then reply

> "Command failed – (F**8**) Command did not address known transactive neighbor connection attributes"

If the system manager connection attribute number does not match a known system manager connection attribute number (e.g., is not a member of {**17**, **28**, **29**, **30**, **31**, and **33**}), or if no ***53*** – *System Manager ID* has been stated as a parameter before this command attempts to configure its attributes, then reply

> "Command failed – (F**9**) Command did not address known system manager connection attributes"

If the asset connection attribute number does not match a known asset connection attribute number (e.g., is not a member of {**6**, **17**, **25**, **28**, **29**, **30**, **31**, **33**, **36**, and **38**}), or if no ***2*** – *Asset ID* has been stated as a parameter before this command attempts to configure its attributes, then reply

> "Command failed – (F**10**) Command did not address known asset connection attributes"

If the local information connection attribute number does not match a known local information connection attribute number (e.g., is not a member of {**17**, **26**, **28**, **29**, **30**, **31**, and **33**}), or if no ***48*** – *Local Information ID* has been stated as a parameter before a command attempts to configure its attributes, then reply

> "Command failed – (F**11**) Command did not address known local information connection attributes"

If the command cannot be completed for any other reason, reply

> "Command failed – (F**6**) Unknown reason"

Otherwise,

Reply, "Command succeeded – (**S1**)"

Finalize any changes to connection attributes that were specified in the file or in-line command.

Run a *Connection Configuration Test* on this connection.

Run a transactive node *Configuration Test* on this transactive node.

Connection Configuration Test()—a simple test of a given connection's attributes to determine if the connection may transition into or remain in its ***2*** – *Configured* state. A connection in either its **3** – Connected or ***4*** – *Lost Connection* state has, by definition passed its *Connection Configuration Test*. If a connection passes its *Connection Configuration Test*, it should be in state **2** – *Configured*; if it fails, it should be in state ***1*** - *Listed*.

A *Connection Configuration Test* is not a system command. It should be initiated by the logic of the transactive node and by the transactive node itself. It should be run for a given connection anytime that the *Configure()* command has run successfully and might have therefore modified the configuration of the connection.

Test Parameters

All = test each connection according to its connection type

TransactiveNeighbor = (***52*** – *Transactive Neighbor ID*)—conduct the test on this transactive neighbor connection.

SystemManager = (***53*** – *System Manager ID*)—conduct the test on this system manager connection.

Asset = (***2*** – *Asset ID*)—conduct the test on this asset connection.

LocalInformation = (***48*** – *Local Information ID*)—conduct the test on this local information connection.

Test Logic

If upon checking attribute ***32*** – *Connection Status* for a connection, this connection is found to be in either state "**3**" (***3*** - *Connected*) or "**4**" (***4*** – *Lost Connection*), then

Test passed - (S**1**) The *Connected* and *Lost Connection* states, by definition, pass the *Connection Configuration Test*

For each configured ***52*** – *Transactive Neighbor ID*, if any of the attributes ***10*** – *Receive TIS Source*, ***11*** – *Receive TFS Source*, ***12*** – *Send TIS Targets*, ***13*** – *Send TFS Targets*, ***32*** – *Connection Status*, or ***29*** – *Connection Partner Type* have not been configured, then

Test failed – (F**1**) Transactive neighbor connection is not configured

Set attribute ***32*** – *Connection Status* = "**1**" (connection state ***1*** – *Listed*) for this connection.

For each configured ***53*** – *System Manager ID*, if either of the attributes ***32*** – *Connection Status* or ***29*** - *Connection Partner Type* have not been configured, then

Test failed – (F**2**) System manager connection is not configured

Set attribute ***32*** – *Connection Status* = "**1**" (connection state ***1*** – *Listed*) for this connection.

For each configured ***2*** – *Asset ID*, if any of the attributes ***6*** – *Toolkit Function*, ***25*** – *Asset Output Targets*, ***32*** – *Connection Status*, or ***29*** – *Connection Partner Type* have not been configured, then

Test failed – (F**3**) Asset connection is not configured

Set ***32*** – *Connection Status* = "**1**" (connection state ***1*** – *Listed*) for this connection.

For each configured ***48*** *– Local Information ID*, if any of the attributes ***26*** *– Local Information Source*, ***32*** *– Connection Status*, or ***29*** *– Connection Partner Type* have not been configured, then

Test failed – (F**4**) Local information connection is not configured

Set ***32*** *– Connection Status* = "**1**" (connection state ***1*** *– Listed*) for this connection.

Otherwise

Test passed – (S**2**)

Set ***32*** *– Connection Status* = "**2**" (connection state **2** *– Configured*) for this connection.

Connect() Command—directs a configured connections to be completed between this transactive node and one of its connection partners.

Command Parameters

Connection = ([All / Connection ID])—identifies one connection that is to be completed from this transactive node to a configured connection with a transactive neighbor, system manager, asset, or local information source. If the parameter "All" is used, the transactive node should attempt to apply the command logic sequentially to every configured connection (e.g., every connection for which a ***52*** *- Transactive Neighbor ID*, ***53*** *– System Manger ID*, ***2*** *– Asset ID*, or ***48*** *– Local Information ID* has been configured).

Command Logic

If the entity that made this command is not the local system manager and if the entity has not been explicitly given permission to make this system management command among the commands in its ***31*** *– Connection Partner's System Management Permissions*, then reply

"Command failed – (F**1**) Permissions do not include this command."

If the *Connection ID* parameter of this command cannot be recognized from among the sets of configured ***52*** - *Transactive Neighbor ID*, ***52*** *– System Manager ID*, ***2*** *– Asset ID*, or ***48*** *– Local Information ID* at this transactive node, then reply

> "Command failed – (**F2**) Connection ID was not recognized from configured connections."

If the entity making this command is not among the ***30*** *– Entities Permitted to modify this Connection* for the referenced connection, then reply

> "Command failed – (**F3**) Entity does not have permission to change this connection."

If upon review of its ***32*** *– Connection Status*, the referenced connection is determined to be in its ***3*** *– Connected* state, then reply

> "Command succeeded – (S**1**) Connection was already completed."

If upon review of its ***32*** *– Connection Status*, the referenced connection is determined to be in its ***1*** *– Listed* state, then reply

> "Command failed – (**F4**) Connection cannot be completed from present connection state."

If the given connection cannot be completed for any other reason, reply

> "Command failed – (**F5**) Unknown reason"

> **If** ***32*** *– Connection Status* = "**3**" (connection status ***3*** *– Connected*), then set ***32*** *– Connection Status* = "**2**" (connection state ***2*** *– Configured*) for the referenced connection.

Otherwise,

Reply, "Command succeeded – (S**2**)"

Complete the referenced connection

Set ***32*** *– Connection Status* = "**3**" (connection state ***3*** *– Connected*) for the referenced connection.

Disconnect() Command—system management command by which a transactive node is asked to disconnect a connection between this transactive node and one of its connection partners.

Command Parameters

Connection = ([All / Connection ID])—identifies one connection that is to be disconnected between this transactive node and a transactive neighbor, system manager, asset, or local information source. If the parameter "All" is used, the transactive node should attempt to apply the command logic sequentially to every configured connection (e.g., every connection for which a ***52*** *- Transactive Neighbor ID*, ***53*** *– System Manger ID*, ***2*** *– Asset ID*, or ***48*** *– Local Information ID* has been configured).

Command Logic

If the entity that made this command is not the local system manager and if the entity has not been explicitly given permission to make this system management command among the commands in its ***31*** *– Connection Partner's System Management Permissions*, then reply

"Command failed – (F**1**) Permissions do not include this command."

If the *Connection ID* parameter of this command cannot be recognized from among the sets of configured ***52*** *- Transactive Neighbor ID*, ***52*** *– System Manager ID*, ***2*** *– Asset ID*, or ***48*** *– Local Information ID* at this transactive node, then reply

"Command failed – (F**2**) Connection ID was not recognized from configured connections."

If the entity making this command is not among the ***30*** – *Entities Permitted to modify this Connection* for the referenced connection, then reply

> "Command failed – (**F3**) Entity does not have permission to change this connection."

If upon review of its ***32*** – *Connection Status*, the referenced connection is determined to be in either its **2** – *Configured* or ***1*** - *Listed* state, then reply

> "Command succeeded – (S**1**) Connection was already disconnected."

If the given connection cannot be completed for any other reason, reply

> "Command failed – (F**4**) Unknown reason"

Otherwise,

> Reply, "Command succeeded – (S**2**)"
>
> Disconnect the referenced connection
>
> Set ***32*** – *Connection Status* = "**2**" (connection state **2** – *Configured*) for the referenced connection.

Loss of Connection Event()—a diagnostic process at this transactive node observes the health and activity of each connection. If the connection should fail, the diagnostic process initiates a *Loss of Connection Event*. This event transitions the respective connection into a temporary *Lost Connection* state, from which the ramifications of the event may be addressed and handled. This transactive node is permitted to remain in its *Operational* state in the meantime, according to the logic of the present state model.

Event Parameters—None.

Said "diagnostic process" should apply to a connection that is in either its ***3*** – *Connected* or ***4*** - *Lost Connection* states. The means by which a connection may be monitored may involve one or more of these suggested mechanisms:

- Observation of interactions with connection partners that occur or fail to occur at times that such interactions were anticipated
- Occasional "pings" of connection partners to determine whether they remain communicative
- A "heartbeat" mechanism that ensures connection partners that a connection remains active. (A "heartbeat" between transactive neighbors should be bidirectional because both transactive neighbors will share this to monitor the connection. Other connection partners may not be transactive nodes, in which case the heartbeat may be unidirectional to satisfy the transactive node)

Event Handler Logic

This logic applies to a connection that is in its **3** – *Connected* state.

If a connection is no longer working based on findings from the diagnostic process,

Set **32** – Connection Status = "**4**" (connection state **4** – Lost Connection) for this connection.

Add a record of the UTC standard time at which the event occurred into the ***55*** – *Lost Connection Event Buffer.*

Start a timer to keep track of how long this connection remains in its *Lost Connection* state.

Re-Establish Connection Event()—a diagnostic process recognizes that a connection has become restored for a connection that was in its *Lost Connection* state. The connection reverts to its *Connected* state.

Event Parameters—None.

This event handler should use the same diagnostic process that was described above for the *Loss of Connection Event.*

Event Handler Logic

This logic applies only to a connection that is in its temporary *Lost Connection* state.

If prior to the occurrence of a *Terminate Connection Event()*, this transactive node recognizes that a lost connection has become restored, then

Set ***32*** – *Connection Status* = "**3**" (connection state ***3*** – *Connected*) for the respective connection

Stop the *Loss of Connection Event* timer.

Re-commence interactions with the respective connection partner via this connection.

Terminate Connection Event()

Event Parameters—None.

This event handler should use the same diagnostic process that was described above for the *Loss of Connection Event* and *Re-Establish Connection Event.*

Event Handler Logic

This logic applies to a connection that is in its *Lost Connection* state.

If the *Loss of Connection Event* timer exceeds ***54*** *– Connection Timeout Period* for this connection, then

Set ***32*** *– Connection Status* = "**2**" (connection state **2** *– Configured*) for this connection

Issue alert, "(A**1**) – Terminate Connection Event occurred by timeout for connection [Connection ID]."

If upon reviewing the contents of the ***55*** *– Loss of Connection Event Buffer* it is observed that the numbers of *Loss of Connection Events* in the last hour has exceeded the criteria in ***56*** *– Allowed Frequency of Loss of Connection Events*, then

Set ***32*** *– Connection Status* = "**2**" (connection state **2** *– Configured*) for this connection

Issue alert, "(A2) – Terminate Connection Event – Too many hourly events for connection [Connection ID]."

If upon reviewing the contents of the ***55*** *– Loss of Connection Event Buffer* the numbers of *Loss of Connection Events* in the last **24** hours has exceeded the criteria in ***56*** *– Allowed Frequency of Loss of Connection Events*, then

Set ***32*** *– Connection Status* = "**2**" (connection state **2** *– Configured*) for this connection

Issue alert, "(A3) – Terminate Connection Event – Too many daily events for connection [Connection ID]."

6.1.15 Connection State Transition Table

Table **20** is the state transition table for the four types of connections that are to be managed by a transactive node. Refer to the diagrammatic representation of the connection state transitions in Figure **30** that should represent these same state transitions.

Table 20. State Transition Table for Connections of Four Types

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
11a	*Connection Configuration Test Failed*	***1*** *- Listed*	Connection attributes **2** – *Asset ID*, ***10*** *– Receive TIS Source*, ***11*** *– Receive TFS Source*, ***12*** *– Sent TIS Targets*, ***13*** *– Send TFS Targets*, ***25*** *– Asset Output Targets*, ***26*** *– Local Information Source*, ***32*** *– Connection Status*, ***29*** *– Connection Partner Type*, ***48*** *– Local Information ID*, **52** – Transactive Neighbor ID, and ***53*** *– System Manager ID*		***1*** *- Listed*	Connection event log entry	Test failed – [(F**1**) Transactive neighbor connection is not configured / (F**2**) System manager connection is not configured / **(F3)** Asset connection is not configured / (F**4**) Local information connection is not configured]	Connection event log entry
11b	*Configure*	***1*** *- Listed*	Source of command; command parameters; *Filename*; lists of configurable attributes (see command definition),	Nearly any connection attribute may be configured. See the command	***1*** *- Listed*	Reply: "Command Succeeded – (S**1**)" Action: Run Connection Configuration Test Action: Run	Success – (S**1**)	Connection command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			connection attributes **2** - *Asset ID*, **30** – *Entities Permitted to Modify this Connection*, **31** - *Connection Partner's System Management Permissions*, **32** - *Connection Status*, **48** - *Local Information ID*, **52** - Transactive Neighbor ID, **53** - *System Manager ID*	definition for details. Lists of configurable attributes may be found in the command definition.		Configuration Test Connection command log entry		
11c	*Disconnect*	*1 - Listed*	Source of command; command parameters; connection attributes **2** - Asset ID, **30** - Entities Permitted to Modify this Connection, **31** - Connection Partner's System Manageme		*1 - Listed*	Reply: "Command succeeded – (S**1**) Connection already disconnected" Connection command log entry	Success – (S**1**) Connection already disconnected	Connection command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			nt Permissions, **32** - Connection Status, **48** - Local Information ID, **52** - Transactive Neighbor ID, **53** - System Manager ID					
11d	*Fail to Configure*	***1*** *- Listed*	Source of command; command parameters; *Filename*; lists of configurable attributes (see command definition), connection attributes ***2*** - *Asset ID*, ***30*** *– Entities Permitted to Modify this Connection*, ***31*** - *Connection Partner's System Management Permissions*, ***32*** - *Connection Status*, ***48*** - *Local Information ID*,		***1*** *- Listed*	Reply: "Command failed – [(F**1**) Permissions do not include this command / (F**2**) Configure command not allowed from Operational state / (F**3**) File cannot be found or opened / (F**7**) Entity making command does not have permission to configure this connection / (F**8**) Command did not address known transactive neighbor	Command failed - [(F1) Permissions do not include this command / (F**2**) Configure command not allowed from Operational state / (F**3**) File cannot be found or opened / (F**7**) Entity making command does not have permission to configure this connection / (F**8**) Command did not address known transactive neighbor connection	Connection command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			52 - Transactive Neighbor ID, ***53*** *- System Manager ID*			connection attributes / (F**9**) Command did not address known system manager connection attributes / (F**10**) Command did not address known asset connection attributes / (F**11**) Command did not address known local information connection attributes / (F**6**) Unknown reason]" Connection command log entry	attributes / (F**9**) Command did not address known system manager connection attributes / (F**10**) Command did not address known asset connection attributes / (F**11**) Command did not address known local information connection attributes / (F**6**) Unknown reason]	
11e	*Fail to Connect*	***1*** *- Listed*	Source of command; command parameters; connection attributes **2** - Asset ID, **30** - Entities Permitted to Modify this Connection, **31** - Connection Partner's System		***1*** *- Listed*	Reply: "Command failed – [(F**1**) Permissions do not include this command / (F**2**) Connection ID was not recognized from configured connections / (F**3**) Entity does not have	Failure – [(F**1**) Permissions do not include this command / (F**2**) Connection ID was not recognized from configured connections / (F**3**) Entity does not have permission to change	Connection command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			Management Permissions, **32** - Connection Status, **48** - Local Information ID, **52** - Transactive Neighbor ID, **53** - System Manager ID			permission to change this connection / (F4) Connection cannot be completed from present connection state]" Connection command log entry	this connection / (F4) Connection cannot be completed from present connection state]	
11f	*Fail to Disconnect*	*1 - Listed*	Source of command; command parameters; connection attributes **2** - Asset ID, **30** - Entities Permitted to Modify this Connection, **31** - Connection Partner's System Management Permissions, **32** - Connection Status, **48** - Local Information ID, **52** - Transactive Neighbor ID, **53** - Syste		*1 - Listed*	Reply: "Command failed – [(F1) Permissions do not include this command / (F2) Connection ID was not recognized from configured connections / (F3) Entity does not have permission to change this connection / (F4) Unknown reason]" Connection command log entry	Failure - [(F1) Permissions do not include this command / (F2) Connection ID was not recognized from configured connections / (F3) Entity does not have permission to change this connection / (F4) Unknown reason]	Connection command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			m Manager ID					
12	*Connection Configuration Test Passed*	**1** - *Listed*	Connection attributes **2** – *Asset ID*, **10** – *Receive TIS Source*, **11** – *Receive TFS Source*, **12** – *Sent TIS Targets*, **13** – *Send TFS Targets*, **25** – *Asset Output Targets*, **26** – *Local Information Source*, **32** – *Connection Status*, **29** – *Connection Partner Type*, **48** – *Local Information ID*, **52** – Transactive Neighbor ID, and **53** – *System Manager ID*	**32** – *Connection Status* = "**2**" (connection state **2** - *Configured*)	**2** - *Configured*	Connection event log entry	Test passed – (**S2**) Normal pass condition	Connection event log entry
21	*Connection Configuration Test Failed*	**2** - *Configured*	Connection attributes **2** – *Asset ID*, **10** – *Receive TIS Source*, **11** – *Receive TFS Source*, **12** – *Sent TIS*		**1** - *Listed*	Connection event log entry	Test failed – [(**F1**) Transactive neighbor connection is not configured / (**F2**) System manager connection	Connection event log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			Targets, ***13*** *– Send TFS Targets*, ***25*** *– Asset Output Targets*, ***26*** *– Local Information Source*, ***32*** *– Connection Status*, ***29*** *– Connection Partner Type*, ***48*** *– Local Information ID*, **52** – Transactive Neighbor ID, and ***53*** *– System Manager ID*				is not configured / (**F3**) Asset connection is not configured / (**F4**) Local information connection is not configured]	
22a	*Configure*	**2** - *Configured*	Source of command; command parameters; *Filename*; lists of configurable attributes (see command definition), connection attributes ***2*** *- Asset ID*, ***30*** *– Entities Permitted to Modify this Connection*, ***31*** *- Connection Partner's System*	Nearly any connection attribute may be configured. See the command definition for details. Lists of configurable attributes may be found in the command definition	**2** - *Configured*	<u>Reply</u>: "Command Succeeded – (S**1**)" <u>Action</u>: Run Connection Configuration Test <u>Action</u>: Run Configuration Test Connection command log entry	Success – (S**1**)	Connection command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			Management Permissions, **32** - *Connection Status*, **48** - *Local Information ID*, **52** - Transactive Neighbor ID, **53** - *System Manager ID*					
22b	*Connection Configuration Test Passed*	**2** - *Configured*	Connection attributes **2** – *Asset ID*, **10** – *Receive TIS Source*, **11** – *Receive TFS Source*, **12** – *Sent TIS Targets*, **13** – *Send TFS Targets*, **25** – *Asset Output Targets*, **26** – *Local Information Source*. **32** – *Connection Status*, **29** – *Connection Partner Type*, **48** – *Local Information ID*, **52** – Transactive Neighbor ID, and **53**		**2** - *Configured*	Connection event log entry	Test passed – (S**2**) Normal pass condition	Connection event log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			– *System Manager ID*					
22c	*Disconnect*	**2** - *Configured*	Source of command; command parameters; connection attributes **2** - Asset ID, **30** - Entities Permitted to Modify this Connection, **31** - Connection Partner's System Management Permissions, **32** - Connection Status, **48** - Local Information ID, **52** - Transactive Neighbor ID, **53** - System Manager ID		**2** - *Configured*	Reply: "Command succeeded – (S**1**) Connection already disconnected" Connection command log entry	Success – (S**1**) Connection already disconnected	Connection command log entry
22d	*Fail to Configure*	**2** - *Configured*	Source of command; command parameters; *Filename*; lists of configurable attributes (see command definition),		**2** - *Configured*	Reply: "Command failed – [(F**1**) Permissions do not include this command / (F**2**) Configure command	Command failed - [(F**1**) Permissions do not include this command / (F**2**) Configure command not allowed	Connection command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			connection attributes **2** - *Asset ID*, **30** – *Entities Permitted to Modify this Connection*, *31 - Connection Partner's System Management Permissions*, **32** - *Connection Status*, **48** - *Local Information ID*, **52** - Transactive Neighbor ID, **53** - *System Manager ID*			not allowed from Operational state / (F**3**) File cannot be found or opened / (F7) Entity making command does not have permission to configure this connection / (F**8**) Command did not address known transactive neighbor connection attributes / (F**9**) Command did not address known system manager connection attributes / (F**10**) Command did not address known asset connection attributes / **(F11)** Command did not address known local information connection attributes / (F**6**) Unknown	from Operational state / (F**3**) File cannot be found or opened / (F**7**) Entity making command does not have permission to configure this connection / (F**8**) Command did not address known transactive neighbor connection attributes / (F**9**) Command did not address known system manager connection attributes / (F**10**) Command did not address known asset connection attributes / (F**11**) Command did not address known local information connection attributes / (F**6**) Unknown reason]	

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
						reason]” Connection command log entry		
22e	*Fail to Connect*	**2** - *Configured*	Source of command; command parameters; connection attributes **2** - Asset ID, **30** - Entities Permitted to Modify this Connection, **31** - Connection Partner's System Management Permissions, **32** - Connection Status, **48** - Local Information ID, **52** - Transactive Neighbor ID, **53** - System Manager ID		**2** - *Configured*	Reply: “Command failed – [(F**1**) Permissions do not include this command / (F2) Connection ID was not recognized from configured connections / (F**3**) Entity does not have permission to change this connection / (F**5**) Unknown reason]” Connection command log entry	Failure – [(F**1**) Permissions do not include this command / (F**2**) Connection ID was not recognized from configured connections / (F**3**) Entity does not have permission to change this connection / (F**5**) Unknown reason]	Connection command log entry
22f	*Fail to Disconnect*	**2** - *Configured*	Source of command; command parameters; connection attributes **2** - Asset ID,		**2** - *Configured*	Reply: “Command failed – [(F**1**) Permissions do not include this command / (F2) Conne	Failure - [(F**1**) Permissions do not include this command / (F**2**) Connection ID was not	Connection command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			30 - Entities Permitted to Modify this Connection, **31** - Connection Partner's System Management Permissions, **32** - Connection Status, **48** - Local Information ID, **52** - Transactive Neighbor ID, **53** - System Manager ID			ction ID was not recognized from configured connections / (F**3**) Entity does not have permission to change this connection / (F**4**) Unknown reason]" Connection command log entry	recognized from configured connections / (F**3**) Entity does not have permission to change this connection / (F**4**) Unknown reason]	
23	*Connect*	***2** - Configured*	Source of command; command parameters; connection attributes **2** - Asset ID, **30** - Entities Permitted to Modify this Connection, **31** - Connection Partner's System Management Permissions, **32** - Conne	*32- Connection Status* = "**3**" (connection state ***3** – Connected*)	***3** - Connected*	<u>Reply</u>: "Command succeeded – (S**2**)" Connection command log entry	Command succeeded – (S**2**) Normal completion	Connection command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			ction Status, **48** - Local Information ID, **52** - Transactive Neighbor ID, **53** - System Manager ID					
32	*Disconnect*	*3 - Connected*	Source of command; command parameters; connection attributes **2** - Asset ID, **30** - Entities Permitted to Modify this Connection, **31** - Connection Partner's System Management Permissions, **32** - Connection Status, **48** - Local Information ID, **52** - Transactive Neighbor ID, **53** - System Manager ID	***32** – Connection Status* = "**2**" (connection state **2** - *Configured*)	***2** - Configured*	<u>Reply</u>: "Command succeeded – (S**2**)" <u>Action</u>: Sever connection to this communication partner Connection command log entry	Success – (S**2**) Normal completion	Connection command log entry
33a	Connection	*3 - Conn*	Connection attributes **2**		*3 - Conn*	Connection event log	Test passed –	Connection event log

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
	Configuration Test Passed	*ected*	*– Asset ID*, ***10*** *– Receive TIS Source*, ***11*** *– Receive TFS Source*, ***12*** *– Sent TIS Targets*, ***13*** *– Send TFS Targets*, ***25*** *– Asset Output Targets*, ***26*** *– Local Information Source*, ***32*** *– Connection Status*, ***29*** *– Connection Partner Type*, ***48*** *– Local Information ID*, **52** – Transactive Neighbor ID, and ***53*** *– System Manager ID*		*ected*	entry	(S1) Connection already completed	entry
33b	Connect	**3** - *Connected*	Source of command; command parameters; connection attributes **2** - Asset ID, **30** - Entities Permitted to Modify this Connection, **31** - Conne		**3** - *Connected*	Reply: "Command succeeded – (S1) Connection already made" Connection command log entry	Command succeeded – (S1) Connection already made	Connection command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			ction Partner's System Management Permissions, **32** - Connection Status, **48** - Local Information ID, **52** - Transactive Neighbor ID, **53** - System Manager ID					
33c	Fail to Configure	**3** - *Connected*	Source of command; command parameters; *Filename*; lists of configurable attributes (see command definition), connection attributes **2** - *Asset ID*, **30** – *Entities Permitted to Modify this Connection*, **31** - *Connection Partner's System Management Permissions*, **32** - *Connection*		**3** - *Connected*	Reply: "Command failed – [(F**1**) Permissions do not include this command / (F**2**) Configure command not allowed from Operational state / (F**12**) Configure command not allowed from connected connection states]" Connection command log entry	Command failed - [(F**1**) Permissions do not include this command / (F**2**) Configure command not allowed from Operational state / (F**12**) Configure command not allowed from connected connection states]	Connection command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			Status, ***48** - Local Information ID*, **52** - Transactive Neighbor ID, ***53** - System Manager ID*					
33d	Fail to Connect	***3** - Connected*	Source of command; command parameters; connection attributes **2** - Asset ID, **30** - Entities Permitted to Modify this Connection, **31** - Connection Partner's System Management Permissions, **32** - Connection Status, **48** - Local Information ID, **52** - Transactive Neighbor ID, **53** - System Manager ID		***3** - Connected*	Reply: "Command failed – [(F**1**) Permissions do not include this command / (F**2**) Connection ID was not recognized from configured connections / (F**3**) Entity does not have permission to change this connection / (F**5**) Unknown reason]" Connection command log entry	Failure – [(F**1**) Permissions do not include this command / (F**2**) Connection ID was not recognized from configured connections / (F**3**) Entity does not have permission to change this connection / (F**5**) Unknown reason]	Connection command log entry
33e	Fail to Disconn	***3** - Connected*	Source of command; command		***3** - Connected*	Reply: "Command failed –	Failure - [(F**1**) Permissions do	Connection command

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
	ect		parameters; connection attributes **2** - Asset ID, **30** - Entities Permitted to Modify this Connection, **31** - Connection Partner's System Management Permissions, **32** - Connection Status, **48** - Local Information ID, **52** - Transactive Neighbor ID, **53** - System Manager ID			[(F1) Permissions do not include this command / (F**2**) Connection ID was not recognized from configured connections / (F3) Entity does not have permission to change this connection / (F**4**) Unknown reason]” Connection command log entry	not include this command / (F**2**) Connection ID was not recognized from configured connections / (F**3**) Entity does not have permission to change this connection / (F**4**) Unknown reason]	log entry
34	Loss of Connection Event	**3** - *Connected*	Diagnostic system information from the system that oversees connections; identity of affected connection; and connection attributes ***18** - Time*, ***32** – Connection Status*	***32** – Connection Status* = “**4**” (connection state ***4** – Lost Connection*), and ***55** – Loss of Connection Event Buffer*	***4** – Lost Connection*	Connection event log entry	Diagnostic system detects that a connection to a connection partner is dead while that connection is in its *Connected* state	Connection event log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
42a	*Terminate Connection Event*	***4** – Lost Connection*	Diagnostic system information from the system that oversees connections; identity of affected connection; and connection attributes ***18** - Time*, ***32** - Connection Status*, ***54** - Connection Timeout Period*, ***55** - Loss of Connection Event Buffer*, ***56** - Allowed Frequency of Loss of Connection Events*	***32** – Connection Status* = "**2**" (connection state ***2** - Configured*)	***2** - Configured*	"Alert – [(A1) Terminate Connection Event occurred by timeout for connection [Connection ID] / (A2) Terminate Connection Event – Too many hourly events for connection [Connection ID] / (A3) Terminate Connection Event – Too many daily events for connection [Connection ID]]" Connection event log entry	[(A**1**) Terminate Connection Event occurred by timeout for connection [Connection ID] / (A**2**) Terminate Connection Event – Too many hourly events for connection [Connection ID] / (A**3**) Terminate Connection Event – Too many daily events for connection [Connection ID]]	Connection event log entry
42b	Disconnect	***4** – Lost Connection*	Source of command; command parameters; connection attributes **2** - Asset ID, **30** - Entities Permitted to Modify this Connection, **31** - Connection	***32** – Connection Status* = "**2**" (connection state ***2** - Configured*)	***2** - Configured*	Reply: "Command succeeded – (S**2**)" Action: Sever connection to this communication partner Connection command log entry	Success – (S**2**) Normal completion	Connection command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			Partner's System Management Permissions, **32** - Connection Status, **48** - Local Information ID, **52** - Transactive Neighbor ID, **53** - System Manager ID					
43a	Connect	*4 – Lost Connection*	Source of command; command parameters; connection attributes **2** - Asset ID, **30** - Entities Permitted to Modify this Connection, **31** - Connection Partner's System Management Permissions, **32** - Connection Status, **48** - Local Information ID, **52** - Transactive Neighbor	*32- Connection Status* = "**3**" (connection state *3 – Connected*)	*3 - Connected*	<u>Reply</u>: "Command succeeded – (S**2**)" Connection command log entry	Command succeeded – (S**2**) Normal completion	Connection command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			ID, **53** - System Manager ID					
43b	Re-Establish Connection Event	***4*** *– Lost Connection*	Diagnostic system information from the system that oversees connections; identity of affected connection; and connection attributes ***18*** *- Time*, ***32*** *- Connection Status, and* ***54*** *- Connection Timeout Period*	***32*** *– Connection Status* = "**3**" (connection state ***3*** *– Connected*)	***3*** *- Connected*	Action: Re-establish interface to respective connection partner. Connection event log entry	Diagnostic system detects that a broken connection to a connection partner has become re-established while that connection is in its *Lost Connection* state	Connection event log entry
44a	Connection Configuration Test Passed	***4*** *– Lost Connection*	Connection attributes **2** *– Asset ID*, ***10*** *– Receive TIS Source*, ***11*** *– Receive TFS Source*, ***12*** *– Sent TIS Targets*, ***13*** *– Send TFS Targets*, ***25*** *– Asset Output Targets*, ***26*** *– Local Information Source*, ***32*** *– Connection Status*, ***29*** *–*		***4*** *– Lost Connection*	Connection event log entry	Test passed – (S**1**) Connection already completed	Connection event log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			Connection Partner Type, **48** – *Local Information ID*, **52** – Transactive Neighbor ID, and **53** – *System Manager ID*					
44b	Fail to Configure	*4 – Lost Connection*	Source of command; command parameters; *Filename*; lists of configurable attributes (see command definition), connection attributes **2** - *Asset ID*, **30** – *Entities Permitted to Modify this Connection*, **31** - *Connection Partner's System Management Permissions*, **32** - *Connection Status*, **48** - *Local Information ID*, **52** - Transactive Neighbor ID,		*4 – Lost Connection*	Reply: "Command failed – [(F**1**) Permissions do not include this command / (F**2**) Configure command not allowed from Operational state / (F**12**) Configure command not allowed from connected connection states]" Connection command log entry	Command failed - [(F**1**) Permissions do not include this command / (F**2**) Configure command not allowed from Operational state / (F**12**) Configure command not allowed from connected connection states]	Connection command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			***53** - System Manager ID*					
44c	Fail to Connect	***4** – Lost Connection*	Source of command; command parameters; connection attributes **2** - Asset ID, **30** - Entities Permitted to Modify this Connection, **31** - Connection Partner's System Management Permissions, **32** - Connection Status, **48** - Local Information ID, **52** - Transactive Neighbor ID, **53** - System Manager ID		***4** – Lost Connection*	Reply: "Command failed – [(F**1**) Permissions do not include this command / (F**2**) Connection ID was not recognized from configured connections / (F3) Entity does not have permission to change this connection / (F**5**) Unknown reason]" Connection command log entry	Failure – [(**F1**) Permissions do not include this command / (F**2**) Connection ID was not recognized from configured connections / (F**3**) Entity does not have permission to change this connection / (**F5**) Unknown reason]	Connection command log entry
44d	Fail to Disconnect	***4** – Lost Connection*	Source of command; command parameters; connection attributes **2** - Asset ID, **30** - Entities Permitted		***4** – Lost Connection*	Reply: "Command failed – [(F1) Permissions do not include this command / (F**2**) Connection ID was not	Failure - [(F**1**) Permissions do not include this command / (F**2**) Connection ID was not recognized from	Connection command log entry

Row	Internal Function	Acts Upon Current State	Using Input	To Set Attributes	Producing		On the Condition	Info. Gathered & Recorded
					Next State	Output		
			to Modify this Connection, **31** - Connection Partner's System Management Permissions, **32** - Connection Status, **48** - Local Information ID, **52** - Transactive Neighbor ID, **53** - System Manager ID			recognized from configured connections / (**F3**) Entity does not have permission to change this connection / (**F4**) Unknown reason]" Connection command log entry	configured connections / (F**3**) Entity does not have permission to change this connection / (F**4**) Unknown reason]	

6.1.16 Log Entries

The state transition tables in this section have consistently indicated outputs to a log table. It will be good practice to create a log entry record for each command and event that is encountered by the transactive node and its connections. Instead of defining each log entry at the point that it was introduced in the state transition tables, it may be preferred to establish practices for the contents of these records based on their types and by whether they affect the transactive state model or that of the transactive node's connections:

1. Command log entry—to be recorded each time a transactive node system management command is received and responded.

   {Source of the command, time received, command ID, command parameters, **5** – *Node Version*, **7** – *Node Status* after the command, completion condition}

2. Connection command log entry—to be recorded each time a connection system management command is received and responded.

   {Source of the command, time received, command ID, target connection ID, ***32*** – *Connection Status* after the command, completion condition}

3. Event log entry—to be recorded each time a transactive node event occurs and is responded to.

   {Event time, event ID, **5** – Node Version, **7** – Node Status after the event, completion condition}

4. Connection event log entry—to be recorded each time a connection event occurs and is responded to.

   {Event time, event ID, target connection ID, ***32*** – *Connection Status* after the event, completion condition}

5. Test log entry—to be recorded each time a transactive node test occurs and is responded to.

   {Test time, test ID, **5** – Node Version, **7** – Node Status after the test, completion condition}

6. Connection test log entry—to be recorded each time a connection test occurs and is responded to.

   {Test time, test ID, target connection ID, ***32*** – *Connection Status* after the test, completion condition

6.1.17 Operational Sub-States Table

The table below represents that state transitions of a transactive node that has been configured, connected and is now in the overall operational state and status. Note that there is no start or final state in this table. All states may be intermediary. Refer to the toolkit framework for the algorithmic framework facilitated by this part of the state model.

Table 21: State Transition Model for Transactive Nodes within an Operational State ("U" = unconditional)

Row	Internal Function	Acts Upon Current State	By Setting Attributes	Using Inputs	Producing		On the Condition	Info. Gathered and Recorded
					Next State	Output		
A1	Receive TIS	Operational (Listening)	**7**	TIS Message	TIS Received		U	**1**, **7**. **18**, Received TIS message
A2	Formulate TIS	Operational (Listening)	**7**	Attribute **23** (TIS Buffer)	TIS Processed	Stop TIS Timer, Outgoing TIS Message(s)	TIS Timer > TIS Timer Max or TIS received from all inputs	**1**, **7**, **18**, Processed TIS message
A3	Receive TFS	Operational (Listening)	**7**	TFS Message	TFS Received		U	**1**, **7**, **18**
A4	TFS	Operational (Listening)	**7**	Attribute **24** (TFS Buffer)	TFS Processed	Stop TFS Timer, Outgoing TFS Messages	TFS Timer > TFS Timer Max or TFS received from all inputs.	(**1**), (**7**), (**18**), Processed TFS message
A5	Update TIS Buffer	TIS Received	**7**, **23**	TIS Message	Operational (Listening)	Start TIS Timer	No TIS Receive Error, and Start TIS Timer if it is not already running.	(**1**), (**7**), (**18**), **23**
A6	Handle Non-fatal TIS Receive Error	TIS Received	**7**	TIS Message	Operational (Listening)	Non-fatal TIS Receive Error	TIS Receive Error	(**1**), (**7**), (**18**)
A6a	Handle Fatal TIS Receive	TIS Received	**7**	TIS Message	Stopped	Fatal TIS Receive Error	Fatal TIS Receive Error	(**1**), (**7**), (**18**)

Row	Internal Function	Acts Upon Current State	By Setting Attributes	Using Inputs	Producing		On the Condition	Info. Gathered and Recorded
					Next State	Output		
	Error							
A7	Send TIS	TIS Processed	**7**	Outgoing TIS Messages	Operational (Listening)	TIS Message(s) to each neighbor	Send TIS if and only if a TIS has not already been sent within the Time Interval	(**1**), (**7**), (**18**), Sent TIS messages
A7a	Send TIS	Operational (Listening)	**7**	Outgoing TIS Messages	Operational (Listening)	TIS Receive Error, TIS Message(s) to each neighbor	Send TIS if and only if any inputs from neighbors have not been received within the time interval	(**1**), (**7**), (**18**), Sent TIS messages
A7b	Handle non-fatal TIS processing error	TIS Processed	**7**	TIS Processing Error	TIS Processed	Recovery	Non-fatal TIS Processing error	(**1**), (**7**), (**18**) Received TIS Message, Generated error
A7c	Handle fatal TIS processing error	TIS Processed	**7**	TIS Processing Error	Stopped		Fatal TIS Processing Error	(**1**), (**7**), (**18**) Received TIS Message, Generated error
A8	Update TFS Buffer	TFS Received	**7**, **24**	TFS Message	Operational (Listening)	Start TFS Timer	No TFS Receive Error, and Start TFS timer if it is not already	(**1**), (**7**), (**18**), **24**

Row	Internal Function	Acts Upon Current State	By Setting Attributes	Using Inputs	Producing		On the Condition	Info. Gathered and Recorded
					Next State	Output		
							running	
A9	Handle Non-fatal TFS Receive Error	TFS Received	**7**	TFS Message	Operational (Listening)	Non-fatal TFS Receive Error	TFS Receive Error	(**1**), (**7**), (**18**)
A9a	Handle Fatal TFS Receive Error	TFS Received	**7**	TFS Message	Stopped	Fatal TFS Receive Error	Fatal TFS Receive Error	(**1**), (**7**), (**18**)
A10	Send TFS	TFS Processed	**7**	Outgoing TFS Messages	Operational (Listening)	TFS Message(s) to each neighbor	Send TFS if and only if a TFS has not already been sent within the time interval.	(**1**), (**7**), (**18**), Sent TFS messages
A10a	Send TFS	Operational (Listening)	**7**	Outgoing TFS Messages	Operational (Listening)	TFS Receive Error, TFS Message(s) to each neighbor	Send TFS if and only if any inputs from our neighbors have not been received within the time interval.	(**1**), (**7**), (**18**), Sent TFS messages
A11	Handle non-fatal TFS processing error	TFS Processed	**7**	TFS Processing Error	TFS Processed	Recovery	Non-fatal TFS Processing error	(**1**), (**7**), (**18**) Received TFS Message, Generated error
A11a	Handle fatal TFS	TFS Process	**7**	TFS Processi	Stopped		Fatal TFS Processi	(**1**), (**7**), (**18**) Received

Row	Internal Function	Acts Upon Current State	By Setting Attributes	Using Inputs	Producing		On the Condition	Info. Gathered and Recorded
					Next State	Output		
	processing error	ed		ng Error			ng Error	TFS Message, Generated error

6.1.18 Transactive Control Signal Propagation

6.1.18.1 Problem Statement

Transactive control signals (transactive incentive signal and transactive feedback signal) carry information related to electrical power supply and demand over a wide area network. The signals traverse a network of transactive control nodes to elicit a desired control action from responsive assets in a timely manner. The end-to-end (from generation to end-user customer) transmission time should be less than **3** minutes assuming a transactive control hierarchy of **15** levels spanning a **1000** mile radius. This translates to roughly **12** seconds (**180/15**) per hop time budget including the link transit time. Note that the transactive incentive signals will start at the bulk generators and continue to end-user customers. The transactive feedback signal will start at the end-use customer and will travel through the transactive control hierarchy towards bulk generation. While the TIS and the TFS are decoupled temporally and loosely coupled functionally in the sense that a TFS generation does not have to get triggered by the arrival of a TIS, the two signals still influence each other since the computation of TIS and TFS considers the forecasted values for each signal.

The timing model can be purely clock-driven or more asynchronously event-driven. For example, in some embodiments, a set of neighboring transactive nodes are configured to exchange transactive values with one another until the transactive values converge with one another to an acceptable degree (e.g., within a designated percentage of one another (such as **5**%, **2**%, **1**%, or any other desired degree of tolerance)). Further, in such even-driven systems, when a change occurs within a transactive node (e.g., due to a change in local conditions), the transactive node can be configured to transmit an updated set of transactive signals when its local transactive signals deviate from the previously transmitted signals by more than a relaxation criterion.

If the system becomes highly synchronized, bursts of signals might tax the system infrastructure. If the system becomes too loosely event-driven and asynchronous, it becomes more difficult to confirm that signals will have been conveyed. There is probably some flexibility allowable between these extremes, and the design in this appendix facilitates some flexibility. Regardless, the timing model should recognize that the "conversation" of these signals necessarily changes during the transition from one update interval to the next because the set of future intervals change during this transition.

Figure **31** is a diagram **3100** showing TIS and TFS generation being decoupled. The processing of TIS and TFS inputs is performed in reference to the basic **5**-minute interval structure that is UTC referenced.

6.1.18.2 Transactive Node Object Model Attributes Summary

A set of ten (and in some embodiments, mandatory), configurable attributes *B1* – *B10* are defined below in Table **22**.

6.1.18.3 One exemplary Approach

1. Transactive control nodes of the Demonstration use time synchronization with a tolerance of **200**ms. This is readily achievable using either NTP or SNTP. The synchronization is useful to align transactive signal intervals as well as ease of correlation of data collection and event logs.

2. Each transactive control node has two transactive signal timers: TIS_TIMER and TFS_TIMER. These timers are started upon receipt of a TIS or TFS respectively and impose a delay to allow for arrival of more signals before processing occurs (**12** second default value).

3. Each transactive control node has two "hold-down" timers: TIS_HOLD_DOWN_TIMER and TFS_HOLD_DOWN_TIMER. These timers lock out additional processing to prevent race conditions in the mesh segment of a network of transactive control nodes. (**30** second default value). The value should be >= TIS_TIMER and TFS_TIMER respectively.

4. Each transactive control node has a transactive signal watchdog timer (WATCHDOG_TIMER), which is configured to fire off every T_period (**300** default value) seconds. It is desirable that the WATCHDOG_TIMER be less than

or equal to the value of the smallest interval (currently **300** seconds) used in the communication of the transactive signals.

5. Upon startup, a transactive control node starts the transactive signal watchdog timer. It is **recommended** that **the watchdog timer be aligned with the transactive signal update intervals**. For example, if the transactive signal intervals are {**6**:**00**,**6**:**05**, **6**:**10**, ...} then the watchdog timer is recommended to also be started at **6**:**00** and fire-off every **300** seconds.

6. When the transactive signal watchdog timer expires, if WATCHDOG_TIMER_SIGNAL_GEN_ALWAYS_ON configuration variable is set to TRUE then the node will send TIS and TFS packets to neighboring transactive control nodes. If WATCHDOG_TIMER_SIGNAL_GEN_ALWAYS_ON is set to FALSE and if no signal driven events have taken place in the last interval then the node sends TIS and TFS packets to neighboring transactive control nodes connected to this node. Then, the node restarts the global timer.

7. When the node receives a TIS packet, it starts the TIS_TIMER (if it is not already started), and stores the TIS packet in the local TIS store. The TIS_TIMER represents a transactive signal collection period to allow the transactive control node to receive all possible signals from its neighbors. (Each transactive node typically knows how many transactive neighbors it has and therefore how many transactive signals it should expect to receive. In deeper topologies, the TIS_TIMER and TFS_TIMER will unlikely achieve the desired effect of collecting all signals prior to calculation because signal path lengths will be dissimilar for various signals that are to be received.)

8. When the node receives a TFS packet, it starts the TFS_TIMER (if it is not already started), stores the TFS packet in the local TFS store. The TFS_TIMER represents a transactive signal collection period to allow the transactive control node to receive all possible signals from its neighbors.

9. When the TIS_TIMER expires, the node performs the transactive control computation using the most recent TIS and TFS information stored in its TIS and TFS stores. (Received TIS and TFS signals will often contribute only a small influence to the newly calculated TIS and TFS at a transactive node.)

10. When TFS_TIMER expires, the node starts performs the transactive control computation using the most recent TIS and TFS information stored in its TIS and TFS stores.

11. When the node finishes TIS signal computation, it clears the store and sends a TIS packet to its neighbors (*In simulations, the processing is represented with a delay of **12** seconds*). The TIS_HOLD_DOWN_TIMER is started. No TIS may be sent again until it expires.

12. When the node finishes the TFS signal computation, it clears the cache and sends a TFS packet to its neighbors (*In simulations, the processing is represented with a delay of **12** seconds*). The TFS_HOLD_DOWN_TIMER is started. No TFS may be sent again until it expires.

13. Since the transactive control is a distributed system, there will be times when transactive control signals arrive during the hold-down timer or outside the TIS/TFS timer data collection periods. TIS and TFS signals also may arrive at different parts of the time interval. When a new transactive control signal is received and the corresponding transactive control signal computation is performed, one may find that the resulting TIS/TFS values show no significant changes to the previously sent values in the same "interval." In this case, the transactive control node is recommended to omit or delay the transmission of a new TIS/TFS value. This added feature allows further reductions of both communications chatter and computational cycles. This behavior is controlled by means of two configuration variables: TIS_SIGNAL_SUPPRESS_IF_NO_CHANGE and TFS_SIGNAL_SUPPRESS_IF_NO_CHANGE. If either one of these variables are set to TRUE, then the node will be perform the check for no change of the corresponding TIS or TFS signals and suppress transmission.

14. One of the primary inputs to the transactive control node is the local conditions input. This section encourages inclusion of triggers for computation and transmission of TIS/TFS based on changes in the local conditions. The criteria for incorporation of local conditions will be decided at a later time.

The timers and the operation for an example TIS embodiment are illustrated in diagram **3200** of Figure **32**. The TFS is handled in a similar manner.

In summary, the following desired behavior is expressed in pseudo code format.

Upon node startup:

- Start WATCHDOG_TIMER

Upon receiving a TIS:

- if (TIS_TIMER is not running) && (TIS_HOLD_DOWN_TIMER is not running) && (!TIS_IN_CALCULATION) { Start TIS_TIMER}
- Store received TIS

Upon receiving a TFS:

- if (TFS_TIMER is not running) && (TFS_HOLD_DOWN_TIMER is not running) && (!TIS_IN_CALCULATION) { Start TFS_TIMER }
- Store received TFS

Upon expiration of TIS_TIMER:

- Stop and clear TIS_TIMER
- Set TIS_IN_CALCULATION==true)
- Compute TIS using most recent stored TIS and TFS.
- If (TIS_SIGNAL_SUPPRESS_IF_NO_CHANGE == FALSE) { Send TIS } else { check for change in values of computed TIS with the previously sent TIS. If change { Send TIS } else { do nothing}
- Set TIS_IN_CALCULATION==false)
- If (TIS is sent) {Start TIS_HOLD_DOWN_TIMER}

Upon expiration of TFS_TIMER:

- Stop and clear TFS_TIMER
- Set TFS_IN_CALCULATION==true)
- Compute TFS using most recent stored TIS and TFS.
- If (TFS_SIGNAL_SUPPRESS_IF_NO_CHANGE == FALSE) { Send TFS } else { check for change in values of computed TFS with the previously sent TFS. If change { Send TFS } else { do nothing} }
- Set TFS_IN_CALCULATION==false)

- If (TFS is sent) {Start TFS_HOLD_DOWN_TIMER}

Upon expiration of TIS_HOLD_DOWN_TIMER:

- Stop and clear TIS_HOLD_DOWN_TIMER
- If (no new TIS) { do nothing }
- If (new TIS)
    - Set TIS_IN_CALCULATION==true)
    - Compute TIS using most recent stored TIS and TFS.
    - If (TIS_SIGNAL_SUPPRESS_IF_NO_CHANGE == FALSE) { Send TIS } else { check for change in values of computed TIS with the previously sent TIS. If change { Send TIS } else { do nothing} }
    - Set TIS_IN_CALCULATION==false)
    - If (TIS is sent) { Start TIS_HOLD_DOWN_TIMER }

Figure **33** is a diagram **3300** illustrating an example where a perpetual exchange of signals might become sustained between two transactive node neighbors.

Upon expiration of TFS_HOLD_DOWN_TIMER:

- Stop and clear TFS_HOLD_DOWN_TIMER
- If (no new TFS) { do nothing }
- If (new TFS)
    - Set TFS_IN_CALCULATION==true)
    - Compute TFS using most recent stored TIS and TFS.
    - If (TFS_SIGNAL_SUPPRESS_IF_NO_CHANGE == FALSE) { Send TFS } else { check for change in values of computed TIS with the previously sent TFS}
    - Set TFS_IN_CALCULATION==false)
    - If change { Send TFS } else { do nothing}
    - If (TFS is sent) { Start TFS_HOLD_DOWN_TIMER }

Upon expiration of the WATCHDOG_TIMER:

- If (WATCHDOG_TIMER_SIGNAL_GEN_ALWAYS_ON == TRUE) {
    - If (local_conditions_change == TRUE) || (TIS/TFS is not computed in this period) {

```
            ▪ Recompute TIS/TFS }
        o Send TIS; Send TFS; Start TIS_HOLD_DOWN_TIMER; Start
          TFS_HOLD_DOWN_TIMER}
    • If (WATCHDOG_TIMER_SIGNAL_GEN_ALWAYS_ON == FALSE) {
        o If (local_conditions_change == FALSE) && (we sent TIS/TFS in the last
          transactive signal interval) {
            ▪ Do nothing }
        o Else {
            ▪ Recompute TIS/TFS
            ▪ Send TIS; Send TFS; Start TIS_HOLD_DOWN_TIMER; Start
              TFS_HOLD_DOWN_TIMER}}
```

Table 22: Dictionary of Exemplary Timing Attributes Recommended at a Transactive Node to Facilitate Exchange of Transactive Signals between Transactive Neighbors

No.	Attribute Name	Description	Role	Type	Format	Range of values
B1	*TIS_TIMER*	Started upon receipt of the first *TIS* in the current update interval (See **9** - *Update Frequency*).	Allows for arrival of more *TIS* signals before processing occurs. Helps retard successive transmissions of *TIS* signals.	Single real number.	-	The value **0** (zero) disables the *TIS_TIMER*. Default value: **12** s.
B2	*TFS_TIMER*	Started upon receipt of the first *TFS* in the current update interval (See **9** - *Update Frequency*).	Allows for arrival of more *TFS* signals before processing occurs. Helps retard successive transmissions of *TFS* signals.	Single real number.	-	The value **0** (zero) disables the *TFS_TIMER*. Default value: **12** s.
B3	*TIS_IN_CALCULATION*	If set, indicates that the transactive node is engaged in recalculating	Certain actions are to be prevented during this time to avoid corrupting calculated	Binary condition flag. Dimensionless.	-	**0** – Not busy calculating *TIS*. **1** – Busy calculating

No.	Attribute Name	Description	Role	Type	Format	Range of values
		its *TIS* value.	signals.			TIS.
B4	*TFS_IN_ CALCULATION*	If set, indicates that the transactive node is engaged in recalculating its *TFS* values.	Certain actions are to be prevented during this time to avoid corrupting calculated signals.	Binary condition flag. Dimensionless.	-	**0** – Not busy calculating *TFS*. **1** – Busy calculating TFS.
B5	*TIS_HOLD_ DOWN_TIMER*	Started upon sending a *TIS*. Successive TIS may not be transmitted by this transactive node until after this timer has expired.	Used to suppress transmission of excessive *TIS* messages.	Single real number. Units: s.	-	Use of **0** (zero) as the value disables this timer. Default: **30** s. Timer duration should be shorter than the update interval indicated by ***9*** – *Update Frequency* attribute.
B**6**	*TFS_HOLD_ DOWN_TIMER*	Started upon sending a *TFS*. Successive *TFS* may not be transmitted by this transactive node until after this timer has expired.	Used to suppress transmission of excessive *TFS* messages.	Single real number. Units: s.	-	Use of **0** (zero) as the value disables this timer. Default: **30** s. Timer duration should be shorter than the update interval indicated by ***9*** – *Update Frequency* attribute.
B7	*WATCHDOG_ TIMER*	An event occurs at this interval duration and	Actions, like the transmission of transactive signals and the	Single real number.	-	Use of **0** (zero) as the value disables this

No.	Attribute Name	Description	Role	Type	Format	Range of values
		is aligned with the transitions from one update interval into the next.' (In some embodiments, the watchdog time is aligned with the update interval, but that need not be the case in general. If the watchdog timer is further specified to occur n seconds (e.g., **15** seconds) *prior* to the start of the next *update interval*, it can be more useful to induce transmission of transactive signals and asset control actions that are relevant for the pending *update interval*.)	sending of asset control recommendations to asset systems, may be configured to occur each time the watchdog timer expires. During testing, the watchdog timer duration may be shortened to speed the rate at which observations may be taken and thereby facilitate testing of a transactive node.	Units: s.		timer (e.g., no action may be induced by a watchdog-timer event. If assigned a non-zero value, this duration should be longer than any of the attributes *B1 – TIS Timer*, *B2 – TFS Timer*, *B5- TIS Hold Down Timer*, or *B6 – TFS Hold Down Timer*. (If this is not the case, then watchdog timer events will occur prior to calculating and transmitting transactive signals, and watchdog event induced actions may accumulate.) Default value: Set equal to the duration of the update interval, which is **300** s for the Demonstration.
B8	*WATCHDOG_*	This attribute specifies whether	If set true, this attribute will cause	Logical condition flag: true /	Logic	**0** - False – transactive signals are

No.	Attribute Name	Description	Role	Type	Format	Range of values
	TIMER_ SIGNAL_GEN_ ALWAYS_ ON	transactive signals are to be transmitted or not when a watchdog timer event occurs.	transactive signals to be transmitted upon the occurrence of a watchdog timer event; if set false, only the corresponding transactive signals that were not sent by this transactive node during the expiring watchdog time interval are to be transmitted. See transactive neighbor connection attributes ***59*** – *TIS Sent Flag* and ***60*** – *TFS Sent Flag* attributes.	false.		sent when the watchdog timer duration expires only if corresponding type of transactive signal was not sent during the expiring watchdog timer duration. **1** - True – the default condition – transmit *TIS* and *TFS* transactive signals upon the expiration of the watchdog timer regardless of whether any transactive signals were transmitted during the watchdog timer duration that is expiring.
B9	*TIS_SIGNAL_ SUPPRESS_IF_ NO_CHANGE*	This attribute controls *TIS* generation. If this attribute is set true, the transactive control node will compare computed signal values to the respective previous correspondin	This attribute and the related attribute ***B10*** can reduce the numbers of redundant transactive signals transmitted by this transactive node. There is little value in sending transactive signals that are	Logical condition flag: true / false.	Logic.	**0** – False – the differences between newly calculated and prior transmitted *TIS* signals are not relevant. **1** – True – default value - the

No.	Attribute Name	Description	Role	Type	Format	Range of values
		g transactive signal sent and will not send another *TIS* if the values show no significant changes. Default value is true.	virtually identical to ones that have already been sent. This attribute works in conjunction with attributes *C1* – *C4* (see Appendix C concerning the relaxation stop criterion) and attribute ***B8***. (In some embodiments, if *B9* or *B10* are true, the respective transactive signals will not be sent unless they are significantly different from the last ones sent, regardless of the condition of flag *B8*.)			difference between a newly calculated and prior transmitted *TIS* should be compared, and the newly calculated *TIS* will be transmitted only if the difference was found to be significant. See Appendix C and attributes *C1* – *C4* for a metric of significance.
B10	*TFS_SIGNAL_SUPPRESS_IF_NO_CHANGE*	This attribute controls *TFS* generation. If this attribute is set true, the transactive control node will compare computed signal values to the respective previous corresponding transactive signal sent and will not send another *TFS* if the values show no significant changes.	This attribute and the related attribute *B9* can reduce the numbers of redundant transactive signals transmitted by this transactive node. There is little value in sending transactive signals that are virtually identical to ones that have already been sent. This attribute works in conjunction with	Logical condition flag: true / false.	Logic.	**0** – False – the differences between newly calculated and prior transmitted *TFS* signals are not relevant. **1** – True – default value - the difference between a newly calculated and prior transmitted *TFS* should be

No.	Attribute Name	Description	Role	Type	Format	Range of values
		Default value is true.	attributes *C1* – *C4* (see Appendix C concerning the relaxation stop criterion) and attribute *B8*.			compared, and the newly calculated *TFS* will be transmitted only if the difference was found to be significant. See Appendix C and attributes *C1* – *C4* for a metric of significance.

6.1.19 Transactive Signal Relaxation Stop Criterion

6.1.19.1 Purpose

In certain embodients, transactive nodes periodically send their transactive signals to their neighbors. The timing of this responsibility has recently been specified and will become included in reference code implementations of the transactive node model algorithm (TNMA). The timing specification references a relaxation stop criterion based upon changes observed between the present signal and the most recent prior signal that has been calculated and sent by this transactive node. If the signals are found to have not changed much, this transactive node should not send its calculated signal again during the present update interval.

The purpose of this section is to state the criterion by which a transactive node may discern whether it should continue to send out its calculated transactive signals or not during the present update interval.

6.1.19.2 Relaxation Stop Criterion

A relaxation stop criterion can be used under the following assumptions:

1. Near-term predictions should be known with accuracy. Prediction inaccuracies and perturbations are somewhat more acceptable far into the future because one will have many opportunities to iterate and improve those distant predictions. Near-term predicted inaccuracies and events may necessitate additional iterations until the system relaxes to a steady, negotiated solution.
2. A prediction error decreases inversely proportional to some constant to the power of the number of iterations. The constant represents the improvement expected from each iteration and will usually range from [1,2+). If the constant is set to the conservative value 1, one expects the error not at all to be improved by iteration. If the number is set to 2, one expects that each successive iteration should halve the error. It is conceivable that over-relaxation solutions could allow for constants larger than 2.
3. The impact of an inaccurate prediction is roughly proportional to the predicted interval's duration.

For each future interval *s*, define error ε_s as the absolute difference between the present estimate of the value $V_s(k)$ and the prior estimate of the value $V_s(k\text{-}1)$.

$$\varepsilon_s = \left|V_s(k) - V_s(k-1)\right| \qquad \text{(Eq. C1)}$$

The criterion should be applied consistently to either the value itself or to a relative representation of the value, which further results in dividing the result in Eq. C1 by the absolute value of $V_s(k)$.

Each error ε_s should be factored by a corresponding weighting factor W_s to account for the impacts of the duration of each future interval s and the number of iterations that may be reasonably performed on the prediction.

$$W_s = \frac{D_s}{\gamma^{(t_s - t_0)/D}} \qquad \text{(Eq. C2)}$$

In Eq. C2, D_s is the time duration of interval *s*, and γ is a constant [1,2+) that represents the effectiveness of each iteration, as was described in bullet #2 above. The term $(t_s\text{-}t_0)/D$ represents the number of iterations that could reasonably be completed if iterations are conducted after every D constant time interval between the the start of the predicted interval t_s and the present time t_0. For example, the system can update its calculations every 5 minutes, so one might naturally expect over **12** opportunities for the solution to iteratively converge every hour.

The overall relaxation stop criterion may then be stated as a constant *E* that is proportional to the sum of all the weighting factors. The proportionality constant *K* represents a conservative "typical" error ε_s that would be deemed acceptable. Some trial and error may occur to select the proportionality constant *K* that will result in an acceptable number of iterations.

The time series has been iterated adequately when the weighted sum of errors are less than the constant *E*, in which case iterations should be halted. If, however, the weighted sum of errors is greater than or equal to the constant *E*, then additional iteration should be conducted until errors satisfy the criterion.

$$\mathrm{E} = K\sum_{S} W_s \overset{?}{>} \sum_{S} W_s \cdot \varepsilon_s \qquad \text{(Eq. C3)}$$

The complete criterion is stated in Eq. **C4**.

$$\mathrm{E} = K \cdot \sum_{s=0}^{S} \frac{D_s}{\gamma^{(t_s - t_0)/D}} \overset{?}{>} \sum_{s=0}^{S} \frac{D_s \cdot \varepsilon_s}{\gamma^{(t_s - t_0)/D}} \qquad \text{(Eq. C4)}$$

An example has been worked through in Appendix A using three different values of constant γ. The example uses a set of intervals from the Demonstration of the type that will be used for its transactive signals. The weighting factors for the series of intervals and at the three example values of constant γ have been plotted in graph **3400** of Figure **34**.

Large gamma (e.g. γ = **2.0**) is shown to discount the importance of error in future predictions more than small values of gamma (e.g., γ = **1.0625**). The jagged curve reflects that long interval durations are weighted more than short ones, which is relevant for the Demonstrations intervals that become successively longer after the **12**th, **32**nd, **50**th, and **54**th intervals. The impact of distant future weightings may become negligibly small.

Figure **34** is a graph **3400** showing weighting factors for a set of Demonstration intervals (IST_0=**0**:**00**) using three different values of constant γ.

Table 23: Example Weighting Factors W_s for a Sample Series of Intervals and for Three Different Gamma Values

Sample	D_s		t_s-t_0	W_s		
(#)	(minutes)		(minutes)	γ = 2	γ = 1.25	γ = 1.0625
0	5	1/0/00 0:00	0	5.00E+00	5.00E+00	5.00E+00
1	5	1/0/00 0:05	5	2.50E+00	4.00E+00	4.71E+00
2	5	1/0/00 0:10	10	1.25E+00	3.20E+00	4.43E+00
3	5	1/0/00 0:15	15	6.25E-01	2.56E+00	4.17E+00
4	5	1/0/00 0:20	20	3.13E-01	2.05E+00	3.92E+00
5	5	1/0/00 0:25	25	1.56E-01	1.64E+00	3.69E+00
6	5	1/0/00 0:30	30	7.81E-02	1.31E+00	3.48E+00
7	5	1/0/00 0:35	35	3.91E-02	1.05E+00	3.27E+00
8	5	1/0/00 0:40	40	1.95E-02	8.39E-01	3.08E+00
9	5	1/0/00 0:45	45	9.77E-03	6.71E-01	2.90E+00
10	5	1/0/00 0:50	50	4.88E-03	5.37E-01	2.73E+00
11	5	1/0/00 0:55	55	2.44E-03	4.29E-01	2.57E+00
12	15	1/0/00 1:00	60	3.66E-03	1.03E+00	7.25E+00
13	15	1/0/00 1:15	75	4.58E-04	5.28E-01	6.04E+00
14	15	1/0/00 1:30	90	5.72E-05	2.70E-01	5.04E+00
15	15	1/0/00 1:45	105	7.15E-06	1.38E-01	4.20E+00
16	15	1/0/00 2:00	120	8.94E-07	7.08E-02	3.50E+00

Sample	D_s		t_s-t_0	W_s		
(#)	(minutes)		(minutes)	γ = 2	γ = 1.25	γ = 1.0625
17	15	1/0/00 2:15	135	1.12E-07	3.63E-02	2.92E+00
18	15	1/0/00 2:30	150	1.40E-08	1.86E-02	2.43E+00
19	15	1/0/00 2:45	165	1.75E-09	9.51E-03	2.03E+00
20	15	1/0/00 3:00	180	2.18E-10	4.87E-03	1.69E+00
21	15	1/0/00 3:15	195	2.73E-11	2.49E-03	1.41E+00
22	15	1/0/00 3:30	210	3.41E-12	1.28E-03	1.18E+00
23	15	1/0/00 3:45	225	4.26E-13	6.53E-04	9.80E-01
24	15	1/0/00 4:00	240	5.33E-14	3.35E-04	8.17E-01
25	15	1/0/00 4:15	255	6.66E-15	1.71E-04	6.81E-01
26	15	1/0/00 4:30	270	8.33E-16	8.77E-05	5.68E-01
27	15	1/0/00 4:45	285	1.04E-16	4.49E-05	4.74E-01
28	15	1/0/00 5:00	300	1.30E-17	2.30E-05	3.95E-01
29	15	1/0/00 5:15	315	1.63E-18	1.18E-05	3.29E-01
30	15	1/0/00 5:30	330	2.03E-19	6.03E-06	2.74E-01
31	15	1/0/00 5:45	345	2.54E-20	3.09E-06	2.29E-01
32	60	1/0/00 6:00	360	1.27E-20	6.32E-06	7.63E-01
33	60	1/0/00 7:00	420	3.10E-24	4.34E-07	3.69E-01
34	60	1/0/00 8:00	480	7.57E-28	2.98E-08	1.78E-01

Sample	D_s		t_s-t_0	W_s		
(#)	(minutes)		(minutes)	γ = 2	γ = 1.25	γ = 1.0625
35	**60**	**1/0/00 9:00**	**540**	**1.85E-31**	**2.05E-09**	**8.60E-02**
36	**60**	**1/0/00 10:00**	**600**	**4.51E-35**	**1.41E-10**	**4.16E-02**
37	**60**	**1/0/00 11:00**	**660**	**1.10E-38**	**9.68E-12**	**2.01E-02**
38	**60**	**1/0/00 12:00**	**720**	**2.69E-42**	**6.65E-13**	**9.70E-03**
39	**60**	**1/0/00 13:00**	**780**	**6.57E-46**	**4.57E-14**	**4.69E-03**
40	**60**	**1/0/00 14:00**	**840**	**1.60E-49**	**3.14E-15**	**2.26E-03**
41	**60**	**1/0/00 15:00**	**900**	**3.92E-53**	**2.16E-16**	**1.09E-03**
42	**60**	**1/0/00 16:00**	**960**	**9.56E-57**	**1.48E-17**	**5.28E-04**
43	**60**	**1/0/00 17:00**	**1020**	**2.33E-60**	**1.02E-18**	**2.55E-04**
44	**60**	**1/0/00 18:00**	**1080**	**5.70E-64**	**7.01E-20**	**1.23E-04**
45	**60**	**1/0/00 19:00**	**1140**	**1.39E-67**	**4.82E-21**	**5.96E-05**
46	**60**	**1/0/00 20:00**	**1200**	**3.40E-71**	**3.31E-22**	**2.88E-05**
47	**60**	**1/0/00 21:00**	**1260**	**8.29E-75**	**2.27E-23**	**1.39E-05**
48	**60**	**1/0/00 22:00**	**1320**	**2.02E-78**	**1.56E-24**	**6.72E-06**
49	**60**	**1/0/00 23:00**	**1380**	**4.94E-82**	**1.07E-25**	**3.25E-06**
50	**360**	**1/1/00 0:00**	**1440**	**7.24E-85**	**4.43E-26**	**9.41E-06**
51	**360**	**1/1/00 6:00**	**1800**	**1.53E-106**	**4.66E-33**	**1.20E-07**
52	**360**	**1/1/00 12:00**	**2160**	**3.25E-128**	**4.91E-40**	**1.52E-09**

Sample	D_s		t_s-t_0	W_s		
(#)	(minutes)		(minutes)	γ = 2	γ = 1.25	γ = 1.0625
53	360	1/1/00 18:00	2520	6.87E-150	5.17E-47	1.93E-11
54	1440	1/2/00 0:00	2880	5.82E-171	2.18E-53	9.84E-13
55	1440	1/3/00 0:00	4320	1.17E-257	2.68E-81	2.57E-20
56	1440	1/4/00 0:00	5760	-	3.30E-109	6.72E-28

6.1.19.3 Additional Transactive Node Attributes where a Relaxation Stop Criterion is Employed

Table 24 specifies four additional transactive node attributes that can be used if a transactive node is to employ the relaxation stop criterion as it has been introduced in this appendix. These attributes can be assumed to be assignable at the transactive-node level. It is conceivable that this criterion (or another) and its attributes may in the future be configured differently for each transactive neighbor connection.

Table 24. Dictionary of the Relaxation Stop Criterion Attributes that may be Configured at a Transactive Node

No.	Attribute Name	Description	Role	Type	Format	Range of values
C1	*Relaxation Stop Criterion Proportionality Threshold—TIS*	This is one of the two parameters that determine whether a calculated *Output TIS* time series has adequately relaxed to a steady solution at this transactive	This parameter represents a maximum allowed average absolute difference between consecutively calculated *Output TIS* members	Single real number. This parameter's units of measure are effectively the same as for the *Output TIS*: $/kWh.	-	Typically, [0, 1). Default value: **0.0005**. Set to **0.0** for maximum iterations. Set to **1** to practically

No.	Attribute Name	Description	Role	Type	Format	Range of values
		node. This parameter is the proportionality constant *K* that is shown in Eq. **C4**.	*TIS*$_n$. The magnitudes of attributes **C1** and **C3** together affect how similar *Output TIS* time series should be for us to stop iterating and again transmitting the *Output TIS*. The magnitude of this parameter affects how many times an *Output TIS* will be sent to transactive neighbors by this transactive node.			eliminate iterations altogether. Empirically set this parameter's value to achieve the desired numbers of *Output TIS* being transmitted from this transactive node.
C2	*Relaxation Stop Criterion Proportionality Threshold—TFS*	This is one of the two parameters that determine whether a calculated *Output TFS* time series has adequately relaxed to a steady solution at this transactive node. This parameter is the proportionality constant *K* that is shown in Eq. **C4**.	This parameter represents a maximum allowed average absolute difference between consecutively calculated *Output TFS* members *TFS*$_n$. The magnitudes of attributes **C2** and **C4** together affect how similar *Output TFS* time series should	Single real number. This parameter's units of measure are effectively the same as for an *Output TFS*: Average kW.	-	Typically, [**0**, **100**,**000**). Default value: **100**. Set to **0**.**0** for maximum iterations. Set to **100**,**000** to practically eliminate iterations altogether. Empirically set this parameter's value to achieve the desired

No.	Attribute Name	Description	Role	Type	Format	Range of values
			be for us to stop iterating and again transmitting the *Output TFS*. The magnitude of this parameter affects how many times an *Output TFS* will be sent to transactive neighbors by this transactive node.			numbers of *Output TFS* being transmitted from this transactive node.
C3	*Relaxation Stop Criterion Gamma Parameter—TIS*	This is one of the two parameters that determine whether a calculated *Output TIS* time series has adequately relaxed to a steady solution at this transactive node. This parameter is the constant Y that is shown in Eq. C4.	This parameter represents the relative impact of a sample's duration and a sample's distance into the future as successive Output TIS values are being compared. The magnitudes of attributes C1 and C3 together affect how similar *Output TIS* time series should be for us to stop iterating and again transmitting the *Output TIS*. The magnitude of this parameter	Single real number. This parameter is dimensionless.	-	Range: [1, 2). Default: 1.0 Empirically set this parameter's value to achieve the desired numbers of *Output TIS* being transmitted from this transactive node.

No.	Attribute Name	Description	Role	Type	Format	Range of values
			affects how many times an *Output TIS* will be sent to transactive neighbors by this transactive node.			
C4	*Relaxation Stop Criterion Gamma Parameter—TFS*	This is one of the two parameters that determine whether a calculated *Output TFS* time series has adequately relaxed to a steady solution at this transactive node. This parameter is the constant Υ that is shown in Eq. C4.	This parameter represents the relative impact of a sample's duration and a sample's distance into the future as successive Output TFS values are being compared. The magnitudes of attributes C2 and C4 together affect how similar *Output TIS* time series should be for us to stop iterating and again transmitting the *Output TIS*. The magnitude of this parameter affects how many times an *Output TIS* will be sent to transactive neighbors by this transactive node.	Single real number. This parameter is dimensionless.	-	Range: [1, 2). Default: **1.0** Empirically set this parameter's value to achieve the desired numbers of *Output TIS* being transmitted from this transactive node.

6.2 Appendix B – Transactive Node Toolkit Framework

6.2.1 Terms

This section will sometimes make reference to the following terms, whose nonlimiting definitions are also given below. These definitions do not necessarily apply in all instances and may vary depending on the context.

elastic load		Within the *toolkit framework*, the change in electrical load that is expected as *responsive asset systems* respond to the *transactive incentive signal (TIS)*. Within the *toolkit framework*, information about *elastic load* will be stored into and available from the **Toolkit Response Function Output Parameter Buffer**.
inelastic load		Electrical load that is not responsive to the *transactive incentive signal (TIS)* at a *transactive node*. In certain implementations, it is recommended that *inelastic load* should also include the predicted load from *responsive asset systems* if they were to not respond to the *TIS*. Within the *toolkit framework*, information about *inelastic load* will be stored into and available from the **Inelastic Load Prediction Buffer**.
input transactive feedback signal	*input TFS*	A *transactive feedback signal (TFS)* that has been received from a *transactive neighbor* as an input to the set of calculations that is to be conducted at a *transactive node* at the updated frequency.
input transactive incentive signal	*input TIS*	A *transactive incentive signal (TIS)* that has been received from a *transactive neighbor* as in input to the set of calculations that is to be conducted at a *transactive node* at the update frequency.

interval start time	*IST*	An attribute of *transactive signals*. The series of future times that define the starting times of members of set of future time intervals. The duration of each interval is defined by the time between two consecutive *interval start times.*
other local conditions	*OLC*	A broad set of information and data that will be inputs into the many functions and processes that is to be performed at *transactive nodes*. This set excludes *transactive signals*.
output transactive feedback signal	*output TFS*	A *transactive feedback signal (TFS)* object output from the calculations that are to be conducted at a *transactive node* at the update interval. A *transactive node* prepares an *output TFS* that predicts the average power to be exchanged with a *transactive neighbor* into the future.
output transactive incentive signal	*output TIS*	A *transactive incentive signal (TIS)* object output from the calculations that are to be conducted at a *transactive node* at the update interval.
responsive asset system		A system within the control of a *transactive node* that will change its consumption or generation in response to the *transactive node's transactive incentive signal (TIS)* and *other local conditions.*
toolkit		The *toolkit framework*, *toolkit function libraries*, the set of *toolkit functions*, and/or associated documentation.
toolkit framework		The general functionality and responsibilities at any *transactive node*. The flow in which high-level and more specific *toolkit functions* are coordinated and accomplished.
toolkit function		An individual functional capability that may be implemented at a *transactive node*. There are two main types of *toolkit functions*—incentive and response.

toolkit function library		A set of *toolkit functions* available to implementers. Implementers select *toolkit functions* from this library that can be instantiated and interoperably applied at their *transactive node*.
toolkit load function		A *toolkit function* inserted into the *toolkit framework* process **8. Calculate *Toolkit Resource and Incentive Function*** that calculates energy and energy cost for a resource and other cost components and incentives that will be used in the formulation of the *transactive incentive signal*.
toolkit resource and incentive function		A *toolkit function* inserted into the *toolkit framework* process **6. Calculate *Toolkit Load Function*** that calculates the predicted *inelastic load* and changes in *elastic load* components of the entire load at a *transactive node*.
transactive control	*TC*	A negotiated form of power grid control that uses price-like incentive and feedback signals.
transactive control and coordination system	*TCS*	A distributed system that employs *transactive control and coordination*.
transactive feedback signal	*TFS*	One of the major *transactive signal*s employed by embodiments of a *transactive control and coordination system*. A *transactive node's* reporting of the expected average power to be transferred between two *transactive neighbors* during intervals over the next several days.
transactive incentive signal	*TIS*	One of the major *transactive signal*s employed by embodiments of a *transactive control and coordination system*. A *transactive node's* reporting of the anticipated delivered cost of electrical power at its location at intervals over the next several days.

transactive signal	*TS*	Either the *transactive incentive signal (TIS)* or *transactive feedback signal (TFS).*
transactive neighbors		*Transactive nodes* that exchange electrical energy between them and therefore also exchange *transactive signals.*
transactive node	*TN*	A defined location of the *transactive control and coordination system* that has agreed to exchange *transactive incentive signals (TIS)* and *transactive feedback signals (TFS)* with its *transactive neighbors.*
transactive node model algorithm	*TNMA*	A module of software where the functionality of *transactive control* is created for a *transactive node.* The *Demonstration* chooses to apply this term to software modules that serve this function.
transactive node object		A formal construct possessing attributes that may be used to define the state of a *transactive node* and the transition between those states.
Transactive node object model	*TNOM*	The model of the states of the *transactive node object* and the functions by which it moves from one state to another. The TNOM includes the model of a transactive node object's configuration.
update frequency		Reciprocal of the *update interval.* The update frequency should be made configurable to support future implementations and testing.
update interval		Relatively short time interval between consecutive updates of the *TIS* and *TFS* at each *transactive node.*

6.2.2 Introduction

A *transactive node* represents a predetermined component or region within an electric power grid at which electrical energy may be generated, consumed, imported, or exported. In principle, the *transactive node* construct will be scalable and similarly applicable to from small, end-use equipment (e.g., a distribution transformer, residential thermostats) to large regions (e.g., the boundary of an electric utility). A *transactive node* includes an agent of sorts (e.g., a computer and its software applications) that orchestrates each *transactive node's* responsibilities to:

1. economically balance energy
2. incentivize energy consumption or generation
3. activate its own responsive generation and load resources
4. exchange both *transactive incentive signals (TIS)* and *transactive feedback signals (TFS)* with each of its neighboring *transactive nodes*.

The two *transactive signals*—the *transactive incentive signal (TIS)* and *transactive feedback signal (TFS)*—reveal the predicted local delivered cost of electric energy and the predicted use of a *TN* to exchange electrical energy with its neighbors, given the value of the *TIS* and other predicted local conditions. (While this document refers often to *pairs* of *TIS* and *TFS* signals, the two signals need not necessarily always be received and sent together and simultaneously. Instead, the signals can be decoupled so that they may be sent and received separately.)

These functional behaviors should be designed into the *transactive control and coordination system*. Depending on its complexity, memberships, and location in its power grid, a *transactive node* may assume all, some, or practically none of the responsibilities to be described in this document. The *toolkit function library* construct is one way to organize and teach the responsibilities of a *TN* to those who would wish to define a *transactive node* and have their *transactive node* enter into an existing *transactive control and coordination system*. The *toolkit library* should not only hasten the adoption and implementation of *transactive control*, but it should also standardize implementations of *transactive control* so that the building blocks components will be more interoperable. The *toolkit library* should be available to implementers who may choose from and learn from others' experiences and practices. The template for *toolkit library functions* anticipates providing reference implementation code with which implementers may jump start their instantiation of similar functions.

The functional responsibilities of a *transactive node* will be described at two levels of the *toolkit*:

1. *Toolkit framework*—the high-level computational structure that provides basic *transactive control* functionality of *transactive nodes* and that calls upon specific *toolkit library functions* to enact the functionality of specific incentives and assets.

2. *Toolkit library functions*—the specific functions that account for resource, enact incentives and plan asset responses at *transactive nodes* where these specific functions have been implemented and are relevant. Applicable *toolkit library functions* are called upon and acted upon within the *toolkit framework.*

6.2.3 Toolkit Framework

The *toolkit framework* is a high-level structure for the inputs, functions, processes, and outputs that define *transactive control* functionality at a *transactive node*. The *toolkit framework* will probably be found to encompass the high-level functional responsibilities of the *transactive node model algorithm (TNMA)* module.

(This document primarily addresses the algorithmic functionality of a *transactive node* and its responsibilities toward management of electrical energy. This document may facilitate, but does not intend to specify, functionality toward system management, timing, and data collection that are better addressed within the *transactive node's object model.*)

Figures **35A** and **35B** are a flowchart **3500** that shows the flow of information during each update interval (e.g., **5**-minute *update interval*) at the rate of the *update frequency*. This is a functional flow, not necessarily a recommendation for how a developer will construct the software program. The blocks in this diagram represent functions and processes. The distinction between “functions” and “processes” may be somewhat subjective, but a process will have been defined to have multiple sub-functions and/or sub-processes. Blocks of Figures **35A** and **35B** shown with bold outlines are processes, known to be composed of at least two sub-functions or processes.

The flow of information in Figures **35A** and **35B** is indicated by solid arrows. Information is processed predominantly downward through the diagram, which makes the diagram useful for understanding functional, sequential interdependencies. Other logical flow control and dependencies are shown by dashed arrows.

Information buffers appear in several of the information flow paths. These buffers are available to be mined by data collection processes and might be made accessible to the system management level. (These buffers, if defined as part of a standard *transactive node* definition, can be used as a point of observability for testing. In addition, the option of preloading the buffers may be useful for testing (especially if only the **5**-minute *update frequency* is available).) The buffers also provide recent information that may be used if any prior function or process should fail to promptly complete its responsibilities or provide its output information. The flow in this diagram has been greatly simplified by the assumption that any buffered historical information is available to be used by any other function or process at this *transactive node*.

As part of its data collection design for *transactive* data, a number of buffers can be used. For example, in the illustrated embodiment in Figures **35A** and **35B**, five buffers are identified, the contents of which compose a sufficient snapshot of the calculations that have been completed by the *toolkit framework* and its *toolkit functions* at a *transactive node*. The five buffers are those into which calculation products are to be sent: **Resource Schedule and Cost Buffer**, **Output TIS Buffer**, **Output TFS Buffer**, **Inelastic Load Prediction Buffer**, and **Elastic Load Prediction Buffer**. The freshest, unique buffer records from these five buffers are specified to be collected after any *transactive signal* has been calculated and sent to a *transactive neighbor*. The sampling of these five buffers is *sufficient* in the sense that the outputs from each *toolkit resource and incentive function* and each *toolkit load function* are revealed, the *TIS* and *TFS transactive signals* that have been transmitted from this transactive node are revealed, and the magnitudes of *transactive signals* that have been received may be inferred, if not perfectly known. Alternatively, signal timing and data collection can be initiated by *changes* that have been detected, not by rigid timers.

The following processes and functions are referenced in Figures **35A** and **35B** and will be described in the next sections. Defined functions, processes, and specially defined inputs and outputs of the functions and processes will be shown in bold font in this document.

1. Receive Transactive Signals

1.1. Read *TIS* and *TFS* from *Transactive neighbor*

1.2. Check Authentication and Security

1.2.1. Interact with System Management (Security)

1.3. Check Validity of *Transactive Signals*

1.3.1. Interact with System Management (Validity)

1.4. Update Input *Transactive Signal* Buffer for this *Transactive neighbor*

2. Calculate New *Transactive Signal* Intervals

2.1. Read Present Time

2.2. Calculate First *Interval Start Time* IST_0

2.3. Calculate 5-Minute *Interval Start Times*

2.4. Calculate 15-Mnute *Interval Start Times*

2.5. Calculate 1-Hour *Interval Start Times*

2.6. Calculate 6-Hour *Interval Start Times*

2.7. Calculate 1-Day *Interval Start Times*

2.8. Calculate Interval Durations from *Interval Start Times*

3. Formulate *TIS*

3.1. Refresh Default *Output TIS*

3.2. Calculate Total Costs of Non-Transactive Energy Generation and Imports

3.3. Calculate Total Cost of Energy Imported from *Transactive nodes*

3.4. Calculate Total Capacity Cost / Incentives

3.5. Calculate Total Infrastructure Cost / Incentive

3.6. Calculate Total Other Cost / Incentive

3.7. Calculate *Output TIS*

3.8. Calibrate / Normalize *TIS*

3.9. Interpolate Intervals Service Functions

4. Formulate *TFS*

4.1. Interpolate Intervals Service Functions

4.2. Predict Net Resource Surplus or Shortage

4.3. Disaggregate Net Resource Surplus or Shortage

4.4. Refresh Default *Output TFS*

5. Sum Total Predicted Load

5.1. Interpolate Intervals Service Functions

5.2. Sum *Inelastic Load*

5.3. Sum Change in *Elastic Load*

5.4. Sum Total Inelastic and Change in *Elastic Load*

5.5. Refresh Predicted Total *Inelastic* and *Elastic Load*

6. Calculate Applicable *Toolkit Load Functions*

6.1. Interpolate Intervals Service Functions

6.2m *Toolkit Load Function*

6.3 Refresh Predicted *Inelastic* and *Elastic Loads*

7. Send Transactive Signals (Defined only functionally at a high level)

8. Calculate Applicable Toolkit Resource and Incentive Functions

9. Control *Responsive Asset Systems* (Defined only functionally at a high level)

10. Sum Total Predicted Resources

10.1. Interpolate Intervals Service Functions

10.2. Sum Total Predicted Resource

10.3. Refresh Predicted Total Resource

11. Control Responsive Resource

The next sections will describe examples of the functions in the above list. The sections below are demarcated by the function numbers set forth in the list above, and are not to be confused with the section numbering used outside of this appendix.

1. Receive *Transactive Signals*

Purpose: —*Transactive signals* are signals to be communicated between *transactive nodes* in a *transactive control and coordination system.* It is through *transactive signals* that *transactive nodes* share their temporal and locational costs and thirsts for electrical energy. *Transactive incentive signals (TIS)* and *transactive feedback signals* (*TFS*) should be received from the *transactive node neighbors* at the *update frequency*, which happens to be once every **5** minutes for the *Demonstration.*

This function includes technical validation of received signals to ensure that they were properly formed and that their values are within acceptable norms. Validation is not yet a high priority, and validation processes probably do not need to be standardized across all *transactive nodes.* If an invalid signal is detected, it should be flagged. Additional actions may be taken to notify or alert targeted system operators and reduce the impacts from potentially misleading signals.

Applicability: This function should be completed by a *transactive node* at least once during an *update interval.* If this function fails, functions and processes of the *toolkit framework* that use an input *transactive signal* should revert to buffered historical signals.

Sub-Functions and Sub-Processes: The following sub-functions are iteratively completed until the input *transactive signals* from *transactive neighbors* have been received.

1.1 Read *TIS* and *TFS* from a *Transactive neighbor*—Function by which the *TIS* and *TFS* from a *transactive neighbor* is to be received. Most generally, the implementation details by which this sub-function is to be accomplished should be negotiated by pairs of *transactive neighbors* that will exchange *transactive signals.*

1.2 Check Authentication and Security—Functional block (or blocks) for signals like *transactive signals* that are to be conveyed through the *transactive control and coordination* *system*. The actual functional implementation details for security functions may differ from one implementation to another, but general descriptions for this block should be documented if they are applicable to any *transactive node*.

1.2.1 Interact with System Management (Security)—Actions that are to be taken if **Check Authentication and Security** function fails to authenticate a transactive signal or detects an insecurity. The input transactive signals are terminated if they cannot be authenticated or if security violations are suspected. Actions may include notifications and alerts that are to be conveyed by the system management layer. Specific actions of this function may differ by implementation.

1.3 Check Validity of *Transactive Signals*—Functional block (or blocks) by which the structure or contents of a *transactive signal* may be tested against expected and reasonable structure and content. Examples of checks on the structure of the signals could include verification of adherence to an XML schema, an expected number of future intervals, or the ordering of a series within the signal. An example of a content check would be verification that a signal's values are between stated maximum and minimum values.

1.3.1 Interact with System Management (Validity)—Actions that are to be taken if the **Check Validity** function fails validate transactive signals. The *input transactive signals* are terminated and not used or stored if they cannot be validated. Actions may include notifications and alerts that are to be conveyed by the system management layer. Specific actions of this function may differ by implementation. General functional aspects for this function that should apply to *transactive nodes* should be documented and implemented. More sophisticated actions may be taken, including reducing the Quality attribute of signals that have questionable validity.

1.4 Update *Input Transactive Signal* Buffer for this *Transactive neighbor*—Received transactive signals are saved into the ***Input Transactive Signal* Buffer**. The buffer may be as simple as a running (or circular) list of transactive signal pairs that have been received from *transactive neighbors*. The most recently received pairs or *transactive signals* from each *transactive neighbor* are most relevant within this buffered data. A much longer buffered history may be used at *transactive nodes* that use trending to predict *transactive neighbors'* responses (e.g., elasticity) or to improve the accuracy of their *transactive signal* predictions over time.

Inputs:

- *Input TIS* from a *transactive neighbor*
- *Input TFS* from a *transactive neighbor*
- List of *transactive neighbors* from which *transactive signals* are expected to be received as should be known by the *transactive node* object and available from the **Node State and Status Buffer**. This information in the **Node State and Status Buffer** can be part of the transactive node configuration and state available within the *transactive node object model*, not temporary "buffer" information as the name might imply.

Outputs:

- Buffered copies of Input *TIS* and *TFS*.
- Copies of Input *TIS* and *TFS* pairs conveyed to a data archive by the data collection system layer.
- System management notifications and alerts upon failed security or validation checks, if such system management functions have been defined and if this transactive node is obligated to interact with a system manager.

Dependencies:

- Outputs of this function are used by **Resource Schedules and Cost Buffer**.
- Outputs of this function are used by **Formulate *TIS***.
- Times at which this *transactive node* is eligible to receive *transactive signals* may be managed or limited by the current state of the *transactive node* object, which status is assumed to be known and available from a **Node State and Status Buffer**.
- A set of *transactive neighbors* should be available from attributes of the *transactive node* object, which are assumed to be known and available from a **Node State and Status Buffer**.

<u>Notes</u>:

- The *TIS* and *TFS* are state objects of the Project-Level Infrastructure (PLI) *transactive control and coordination system*. This process expects and checks that *transactive signals* are being received with the specified content and structure, which may be further enforced through the use of, for example, accepted XML schema.

- Considerable tolerance should be built into this function to coordinate with neighboring *transactive nodes* and their readiness to release their transactive signals. The function should be tolerant for when transactive signals are not received, or are not received early enough to influence the present update iteration.

- When an incomplete set of *transactive signals* is received by a *transactive node*, the *transactive node* should rely upon buffered historical information from previous iterations. Unless the power grid's predicted future has changed dramatically, the buffered signals will remain good predictions until input transactive signals are received.

- A **Node State and Status Buffer** has been established within the *toolkit framework* to ensure that it has information it may used concerning timing, *transactive neighbors*, and other status information concerning activities of the *transactive node object*.

- This function interacts with cyber security subsystems. It is assumed here that authentication and other cyber-security tests have been conducted during signal transport or upon signal receipt.

- This function potentially interacts with system management if invalid signals are detected or if notifications or alerts should be conveyed through the system for any reason concerning signals that have been, or should have been, received.

- This function potentially depends upon assumptions and functionality within the state transition diagrams of the *transactive node* state, which design is presently incomplete. It has been assumed that the *transactive node* state diagram has provided states where *toolkit framework* functionality may (or may not) be conducted. Otherwise, it has been assumed that that design of the *transactive node* state transitions does not encroach on the functional responsibilities of the *toolkit framework*.

Figure **36** is a flowchart **3600** illustrating an exemplary receive *transactive incentive signal* process.

2. Calculate New Transactive Signal Intervals

Purpose: Calculate the new *interval start time (IST)* time series that are attributes of the two *transactive signal* object types that are to be formulated and conveyed throughout the *transactive control and coordination system*. See SubAppendix A: Interval Start Time Series Definition for details about an example IST time series and how the series is calculated.

Applicability: This function should be completed by *transactive nodes* at the *update frequency*. In particular implementations, an *update frequency* of once every **5** minutes is used, though other intervals can be used.

Sub-Functions and Sub-Processes: The sub-function steps will be described along with this introduction to the sub-functions. Refer to SubAppendix A for additional details and examples.

2.1 Read Present Time—the present time is locally maintained at each *transactive node* and should be read near the beginning of each iteration. The present time and representations of time are to be maintained using the UTS standard.

2.2 Calculate First Interval Start Time IST_0— to calculate IST_0, round the present time up to the nearest **5**-minute interval.

2.3 Calculate 5-Minute Intervals Start Times— to calculate IST_2 through IST_{11}, add **5** minutes to the prior *IST*.

2.4 Calculate 15-Minute Interval Start Times— to calculate IST_{12}, add **15** minutes to the prior IST_{11}, and round down to a **15**-minute interval. To calculate the remaining **15**-minute intervals IST_{13} through IST_{31}, add **15** minutes to the prior *IST*.

2.5 Calculate 1-Hour Interval Start Times— to calculate IST_{32}, add **1** hour to the prior IST_{31}, and round down to a **1**-hour interval. To calculate the remaining **1**-hour intervals IST_{33} through IST_{49}, add **1** hour to the prior *IST*.

2.6 Calculate 6-Hour Interval Start Times— to calculate IST_{50}, add **6** hours to the prior IST_{49}, and round down to a **6**-hour interval. To calculate the remaining **6**-hour intervals IST_{51} through IST_{53}, add **6** hours to the prior *IST*.

2.7 Calculate 1-Day Interval Start Times— to calculate *IST_{54}*, add 1 day to the prior *IST_{53}*, and round down to a 1-day interval. To calculate the remaining 1-day interval IST_{55}, add 1 day to the prior *IST_{54}*. In certain embodiments, a final IST_{56} can be appended that will unambiguously define the duration of the final interval. (The final IST does not define a new interval, it simply states the end of the last interval.)

2.8 Calculate Interval Durations from Interval Start Times— the function by which *IST* interval durations may be discerned from an *IST* time series is as follows:

2.8.1 Calculate Δt_0_Subtract $IST_1 - IST_0$ to learn the duration of interval Δt_0 that starts at IST_0.

2.8.2 Tentatively Assign Remaining Δt_n—successively subtract $IST_n - IST_{n-1}$ to tentatively assign durations Δt_n. The duration of Δt_{55} has been made unambiguous by appending IST_{56}, which is the end of the last interval.

2.8.3 Perform Checks—certain checks may be possible on the structure of the tentative set of *IST* intervals. In this formulation, both the *IST* times and interval durations should increase or stay the same as one progresses through the series. The tentative set of intervals should be corrected if it does not pass these local checks. The system management layer may be employed to flag, alert, or announce failed checks, but it is the each local node's responsibility alone to produce and use a correct and accurate set of *IST* intervals.

Inputs:

- Present time (determines the first *interval start time* IST_0 for the new *output transactive signals*)

Outputs:

- *IST* time series—Series of *interval start times* $\{IST_0, IST_1, \ldots, IST_N\}$ to be used in *output TIS* and *output TFS* stored into and available from the **Current *IST* Series Buffer**
- Series of *IST* interval durations $\{\Delta t_0, \Delta t_1, \ldots, \Delta t_N\}$ that correspond to the $N+1$ members of the *IST* series stored into and available from the **Current *IST* Series Buffer.**

Function/Process: The process steps were described above as the sub-functions were being introduced. Refer to SubAppendix A for further details, pseudo code, and examples.

Dependencies:

- The function's output is used by process **3. Formulate *TIS***
- The function's output is used by process **4. Formulate *TFS***

Notes:

- The need for synchronicity is low or does not exist in a *transactive control and coordination system*. Therefore, local time should be accurate only to within several tens of seconds. This goal should not be particularly challenging to meet. Regardless, the Demonstration has imposed requirements for and means to achieve impressive synchronicity across its system.
- The *IST* series is an attribute of both the *TIS* and *TFS* state objects.
- While the current *interval start time (IST)* time and interval series are most relevant to the formulation of *transactive signals*, many *toolkit framework* and *toolkit library* functions use access to the current *IST* time and interval series. The **Current *IST* Series Buffer** construct was created to make this accessibility explicit within the *toolkit framework*.

Figure **37** is a flowchart **3700** for an exemplary calculate new *transactive signal* intervals process.

3. Formulate *TIS*

Purpose: Process by which the *TIS*, one of the two *transactive signals*, is to be formulated at a *transactive node*. From its predecessors, this process receives parametric information that is used to determine how energy, capacity, infrastructure, and other influences are to be valued during formulation of the *output TIS* at this *transactive node*.

Applicability: This process should be completed at the *update frequency* by *transactive nodes*. Some of the sub-functions and sub-processes within this process may be trivial or empty at *transactive nodes* where the sub-functions or sub-processes are not needed.

Sub-Functions and Sub-processes:

3.1 Refresh Default *Output TIS*—simply retrieve the most recent *output TIS* from the ***Output TIS* Buffer** at this *transactive node* and refresh its time intervals by submitting it to function **3.10 Interpolate Intervals Service Functions**. The resulting *output TIS* then returned to the ***Output TIS* Buffer** to be used by default if for any reason this *transactive node* does not compute a more current *output TIS* by the time it is used. This sub-function should be completed early during each duration. This potentially creates a race condition in software unless the update status of the buffer is maintained. Thus, in some embodiments, this should be used as a default value

3.2 Calculate Total Cost of Non-transactive Energy Generation and Imports—for each *IST interval*, sum the cost of imported and generated energy from sources that are not *transactive neighbors* at this *transactive node*. Examples include the costs of energy that is imported into the region from Canada, California, or other entities that are not participating in transactive control. Another example would be bulk generation from a gas generator that is dispatched in ways that are not affected by the region's transactive control and coordination system. The data that feed into this function will come from resource schedules and Incentive Toolkit Functions that are employed at this *transactive node*. This function becomes trivial and should not be used at *transactive nodes* that have neither non-transactive imports nor bulk generation.

The output from this function is the sum of products of pairs of energy costs $C_{E,a,n}$ (units: cost per energy) and average generated or imported power $\hat{P}_{G,a,n}$ (units: average power), weighted by the corresponding *IST* interval duration Δt_n (units: time).

$$\sum_{a=1}^{A} C_{E,a,n} \cdot \hat{P}_{G,a,n} \cdot \Delta t_n \qquad \text{(Sub-Function 3.2)}$$

3.3 Calculate Total Cost of Energy Imported from *Transactive nodes*—for each *IST* interval, sum the cost of energy that is predicted to be imported from *transactive neighbors*. At times when energy is to be imported from *transactive neighbors*, the TIS & TFS from those *transactive neighbors* should be treated as special cases of imported energy and treated similarly to non-transactive imported energy (e.g., they result in (C_E, P_G) pairs). The cost of energy from a *transactive neighbor* is that neighbor's *TIS*. The predicted energy to be imported from that neighbor is the neighbor's *TFS* at the boundary between that and this *transactive node.* Exported energy to *transactive neighbors* should be disregarded in the calculation of the *TIS*. (In some embodiments, information about *exported* energy is found in the **Resource Schedules and Cost Buffer**. In such embodiments, Functions **3.2** and **3.3** can filter the buffer contents to address only imported energy, in which case the **Resource Schedules and Cost Buffer** is a complete rich source of information for data collection concerning the outputs of *Toolkit Resource and Incentive Functions* that are being employed at this *transactive node*.) It is conceivable that a *transactive node* could import no energy from its *transactive neighbors*, but the *TFS* shared with the neighbors should be checked nonetheless. (The prediction of energy to be exchanged to or from a transactive neighbor can be predicted by both neighbors, by one of the neighbors, or some other combination.)

As for sub-function **3.2**, the output from this function will continue the sum of products of pairs of energy costs $C_{E,a,n}$ (TIS) (units: cost per energy) and average generated or imported power $\hat{P}_{G,a,n}$ (TFS) (units: average power), weighted by the corresponding *IST* interval duration Δt_n (units: time).

$$\sum_{a=1}^{A} C_{E,a,n} \cdot \hat{P}_{G,a,n} \cdot \Delta t_n \qquad \text{(Sub-Function 3.3)}$$

3.4 Calculate Total Capacity Cost / Incentive—for each *IST* interval, sum the costs that are functions of a capacity. Constraints and demand charges are examples. These are expected to be very non-linear, but they will nonetheless be represented by a capacity cost and the capacity to which they apply. This function may be trivial or empty at *transactive nodes* where no capacity costs or incentives are to be included in the *output TIS*.

The output from this sub-function is the sum of products of pairs of capacity costs $C_{C,b,n}$ (units: cost per power capacity) and average power capacity $\hat{P}_{C,b,n}$ (units: average power) for each respective *IST* interval *n*.

$$\sum_{b=1}^{B} C_{C,b,n} \cdot \hat{P}_{C,b,n} \qquad \text{(Sub-Function 3.4)}$$

3.5 Calculate Total Infrastructure Cost / Incentive—for each *IST* interval, sum the infrastructure (e.g., time-based) costs that should be applied during the interval. This function may be trivial or empty at *transactive nodes* where no infrastructure costs or incentives are to be included in the *output TIS*.

The output from this sub-function is the sum of products of pairs of infrastructure costs $C_{I,c,n}$ (units: cost per time) and the respective interval duration Δt_n (units: time).

$$\sum_{c=1}^{C} C_{I,c,n} \cdot \Delta t_n \qquad \text{(Sub-Function 3.5)}$$

3.6 Calculate Total Other Cost / Incentive—for each *IST* interval, sum those influences that cannot be described by the energy, capacity, and infrastructure functions. (Other Cost / Incentive functions are desirably used infrequently for influences that cannot be described with the other functions. The representation of cost by this function should still be a defensible cost of delivered energy and will be subject to comparison against other cost accountings over relatively long time periods.) This function may be trivial or empty at *transactive nodes* where no other costs or incentives are to be included in the **Output TIS**.

The output from this sub-function is the sum of "Other" costs $C_{O,d,n}$ (units: cost).

$$\sum_{d=1}^{D} C_{O,d,n} \qquad \text{(Sub-Function 3.6)}$$

3.7 Calculate Output TIS—a simple parametric function that combines outputs from above functions to complete calculation of the **Output TIS** for this *transactive node*. The sums completed by five other sub-functions appear in this sub-function. Details about this function are expanded upon in the Section 3.7 Details about the Calculate Output TIS Function.

$$TIS_n = \frac{\sum_{a=1}^{A} C_{E,a,n} \cdot \hat{P}_{G,a,n} \cdot \Delta t_n + \sum_{b=1}^{B} C_{C,b,n} \cdot \hat{P}_{C,b,n} + \sum_{c=1}^{C} C_{I,c,n} \cdot \Delta t_n + \sum_{d=1}^{D} C_{O,d,n}}{\sum_{a=1}^{A} \hat{P}_{G,a,n} \cdot \Delta t_n} \qquad \text{(Sub-Function 3.7)}$$

3.8 Calibrate / Normalize TIS—algorithm by which the output *TIS* are to be compared against and perhaps made to track other cost accounting methods. If the calculation of a *TIS* is meaningful as the delivered cost of electrical energy, it should track other reasonable accountings of the delivered cost of electrical energy over relatively long periods of time. In some embodiments, this is a general requirement on the *TIS*. This general requirement may be enforced by a bias input that will force the TIS to track other less dynamic accountings and thereby correct the *TIS*.

3.9 Interpolate Intervals Service Functions—parse energy and costs from coarse intervals into multiple sub-intervals. This function is necessary because the set of *IST* intervals to be used by the *output TIS* will have divided some prior intervals into sub-intervals. This function is a service function that is called as often as it is desired. The objects *TIS* and *TFS* may simply be replicated for each sub-interval. (While many complex methods may evolve to interpolate and assign costs and average power to sub-intervals, in cetain embodiments of the disclosed technology, the cost and average power from an interval are assigned to its sub-intervals.)

Inputs:

- Energy cost, scheduled / committed non-transactive energy pairings for each non-transactive generation or import resource at a time interval

$$(IST_n^*, \Delta t_n^*, (C_{E,1,n}, \hat{P}_{G,1,n}), (C_{E,2,n}, \hat{P}_{G,2,n}), \ldots, (C_{E,a,n}, \hat{P}_{G,a,n}), \ldots, (C_{E,A,n}, \hat{P}_{G,A,n})),$$

where *n* is a time interval of the *TIS* numbered from **0** to **55**; IST_n is *interval start time n* in a series of *interval start times*; Δt_n is the duration of interval *n*; $C_{E,a,n}$ is the energy cost term (e.g., units $/kWh, like the *TIS*) of the scheduled generation or import resource *a* for *IST* interval *n*, and $\hat{P}_{G,a,n}$ is the average generated or imported power from generation or import resource *a* during time interval *n*. Its units are the same as for *TFS* (e.g., average power). (The asterisk indicates that this series of Interval Start Times and durations will likely differ from those that have been calculated to be used with the Output TIS and Output TFS. The function **3.10 Interpolate Intervals Service** will sort this out for the inputs into the other sub-functions. See, e.g., Figure **C-4**.)

- *Input TIS* and *input TFS* pairings from each *transactive node* neighbor for each time interval

$$(IST_n^*, \Delta t_n^*, (TIS_{1,n}, TFS_{1,n}), (TIS_{2,n}, TFS_{2,n}), \ldots, (TIS_{j,n}, TFS_{j,n}), \ldots, (TIS_{J,n}, TFS_{J,n})),$$

where $TIS_{j,n}$ and $TFS_{j,n}$ are the input transactive signals from *transactive node* neighbor *j* during time interval *n*. This input should be considered a special case of the input described in the preceding bullet. (At times that energy is predicted to be imported from a transactive neighbor, the corresponding *TIS* and *TFS* are special cases of $C_{E,a,n}$ and $P_{G,a,n}$ and will be treated very much the same.)

- *Interval start time* series

$$\{IST_0, IST_1, \ldots, IST_N\}$$

and interval duration series

$$\{\Delta t_0, \Delta t_1, \ldots, \Delta t_N\}$$

to be used for *Output TIS* and *Output TFS*. (These notations do not have asterisks because they are are final intervals to be used in the output *transactive signals* after this iteration.) In Figure **4**, the *Interval Start Time Series* is shown as an input to the function **3.10 Interpolate Intervals Service**, which have the responsibility to resolve any discrepancies between various representations of intervals.

- Energy term(s) C_E from applicable incentive *toolkit functions*, if any. (Energy terms C_E have the same usage and interpretation regardless of whether they are used inside or outside a Toolkit Incentive Function. This term accounts for costs that are roughly proportional to an amount of energy that is being generated or imported into a *transactive node*'s boundary.) The format should be identical to that stated above for non-transactive energy pairings.

- Average Power terms(s) $\hat{P}_G$ from applicable incentive *toolkit functions*, if any. (The average power terms are used similarly regardless of whether they are used in or outside a Toolkit Incentive Function. These terms are an accounting of the average power that is either generated within our imported into a *transactive node* boundary.) The format should be identical to that stated above for non-transactive energy pairings.

- Capacity term(s) C_C from applicable Incentive *toolkit functions*, if any, applicable at each *IST* interval

$$(IST_n^*, \Delta t_n^*, (C_{C,1,n}, \hat{P}_{C,1,n}), (C_{C,2,n}, \hat{P}_{C,2,n}), \ldots, (C_{C,b,n}, \hat{P}_{C,b,n}), \ldots, (C_{C,B,n}, \hat{P}_{C,B,n})),$$

where $C_{C,b,n}$ is the cost to be applied to capacity cost item *b* paired with the capacity $\hat{P}_{C,b,n}$ to which it applies, and $\hat{P}_{C,b,n}$ is the average power capacity for capacity cost item *b* to be multiplied by capacity cost $C_{C,b,n}$ for *IST* interval *n*.

- Infrastructure term(s) C_I from applicable incentive *toolkit functions*, if any

$$(IST_n^*, \Delta t_n^*, C_{I,1,n}, C_{I,2,n}, \ldots, C_{I,c,n}, \ldots, C_{I,C,n}),$$

where $C_{I,c,n}$ is the infrastructure term *c* for the *IST* interval *n*.

- Other term(s) C_O from applicable Incentive *toolkit functions*, if any, for each *IST* interval

$$(IST_n^*, \Delta t_n^*, C_{O,1,n}, C_{O,2,n}, \ldots, C_{O,d,n}, \ldots, C_{I,D,n}),$$

where $C_{O,d,n}$ is the "Other" influence term *d* for *IST* interval *n*.

- Exemplary alternative cost accounting(s) for use by function **3.9 Calibrate / Normalize TIS**. Examples include wholesale energy costs for the same energy or utility expenses.

Interim Calculation Products:

- Total Cost of Non-transactive Energy Imports
- Total Cost of Non-transactive Energy Generation
- Total Cost of Energy Imported from *Transactive neighbors*
- Total Capacity Cost / Incentive
- Total Infrastructure Cost / Incentive
- Total Other Cost / Incentive
- Total Cost
- Total Energy Imported or Generated

- Additionally, interim calculations may be used to represent prior interval information in terms of the new *IST* time series and interval durations that are to be used by the *Output TIS*.

Outputs:

- New "Updated" *Output TIS* at this *transactive node*.

Function/Process: Each of the sub-functions / sub-processes should be defined, but sub-function **3.8 Calculate *Output TIS*** defines the parametric calculation of the *output TIS* from the energy, capacity, infrastructure, and other parameters and how the parameters are to be applied. The implementer who understands sub-function **3.8 Calculate *Output TIS*** will have the insight to formulate toolkit functions and will have considerable flexibility in the way such toolkit functions are formulated.

Dependencies:

- Uses input of new *IST* time series from process **2. Calculate New Transactive Signal Intervals**
- Uses input of *TIS* and *TFS* from at least one *transactive neighbor* via process **1. Receive Transactive Signals**
- Process inputs may come from **Calculate Applicable Toolkit Incentive Functions**
- Process inputs may come from **Resource Schedules and Cost Buffer**.
- *Output TIS* from this process is used by process **7. Send Transactive Signals**.
- *Output TIS* from this process may be used by **Calculate Applicable Toolkit Response Functions dP(TIS,OLC)** if this *transactive node* owns responsive assets.

Notes:

- Each *transactive node* produces one and only one *TIS* for itself for each **5**-minute update iteration.

- The *TIS* itself is a time series that expresses the delivered cost of energy into the future about **3** days, or so, as is defined by the *IST* time series.

Figure **38** is a flowchart **3800** illustrating an exemplary formulate *TIS* process.

Details about the Function 3.7 Calculate Output TIS

Purpose: Describes the final parametric calculation of the *output TIS*. This sub-function consists of a simply stated function of the sum products of other sub-functions **3.2** through **3.7**. This sub-function creates a level of standardization that will help ensure that the TIS at distributed points in a *transactive control and coordination system* are defensible representations of the "delivered cost of energy."

Applicability: A sub-function of **3. Formulate *TIS* Process** that should be calculated at the update frequency at *transactive nodes.*

Sub-Functions and Sub-processes: None. This is a simple arithmetic function of sums that have been calculated by sub-functions **3.2** through **3.7**.

Inputs:

- Summed cost of energy terms

$$\sum_{a=1}^{A} C_{E,a,n} \cdot \hat{P}_{G,a,n} \cdot \Delta t_n \qquad \text{(Sub-Functions 3.2 and 3.3)}$$

from sub-functions **3.2 Calculate Total Cost of Non-Transactive Energy Generation and Imports** and **3.3 Calculate Total Cost of Energy Imported from *Transactive nodes***

- Summed cost of capacity terms

$$\sum_{b=1}^{B} C_{C,b,n} \cdot \hat{P}_{C,b,n} \qquad \text{(Sub-Function 3.4)}$$

from sub-function **3.4 Calculate Total Capacity Cost / Incentive**

- Summed cost of infrastructure terms

$$\sum_{c=1}^{C} C_{I,c,n} \cdot \Delta t_n \qquad \text{(Sub-Function } \mathbf{3.5}\text{)}$$

from sub-function **3.5 Calculate Total Infrastructure Cost / Incentive**

- Summed other costs

$$\sum_{d=1}^{D} C_{O,d,n} \qquad \text{(Sub-Function } \mathbf{3.6}\text{)}$$

from sub-function **3.6 Calculate Total Other Cost / Incentive**

- Summed energy

$$\sum_{a=1}^{A} \hat{P}_{G,a,n} \cdot \Delta t_n \qquad \text{(Function } \mathbf{10}\text{)}$$

that is predicted to be imported and/or generated at this *transactive node* as has been calculated in function **10. Sum Total Predicted Resource.**

<u>Outputs</u>:

- One current *output TIS* time series for this *transactive node*

<u>Function / Process</u>:

This sub-function simply adds the individual cost summations from sub-functions **3.2**, **3.3**, **3.4**, **3.5**, and **3.6** and divides that sum by the total energy that is imported into or generated within the boundaries of this *transactive node* as was summed by sub-function **3.7**:

$$\frac{\sum_{a=1}^{A} C_{E,a,n} \cdot \hat{P}_{G,a,n} \cdot \Delta t_n + \sum_{b=1}^{B} C_{C,b,n} \cdot \hat{P}_{C,b,n} + \sum_{c=1}^{C} C_{I,c,n} \cdot \Delta t_n + \sum_{d=1}^{D} C_{O,d,n}}{\sum_{a=1}^{A} \hat{P}_{G,a,n} \cdot \Delta t_n}, \text{ or} \qquad \text{(Sub-Function } \mathbf{3.7}\text{)}$$

$$TIS = \frac{\underline{energy\ cost + capacity\ cost + infrastructure\ cost + other\ costs}}{Energy}$$

The function shown above for interval *n* should be performed for all intervals that are to be used by the *Demonstration* for its *transactive signals.*

Dependencies:

- Will use sub-function **3.10 Interpolate Intervals Service Functions** to convert intervals of inputs into those of the updated *IST* time series that is to be used by the *output TIS.*
- The output *TIS* produced by this sub-function is one of the two transactive signals that function **7. Send *Transactive Signals*** will act upon and send.

Notes:

- This function assumes that intervals have been aligned and modified to be consistent with the new *IST* intervals that were determined by process **2. Calculate New *Transactive Signal* Intervals**. If that is not the case, the sub-function **3.10 Interpolate Intervals Service Functions** should be applied until inputs to this sub-function have been stated in terms of the *IST* intervals for which the *output TIS* will be produced.
- If properly formulated, the units of *TIS* will be cost per energy. Dimensional unit analysis is a candidate component for conformance testing to be performed on any implementation that follows this *toolkit framework.*

4. Formulate TFS

Purpose: Formulate one current *transactive feedback signal (TFS)* for the electrical interface between this *transactive node* and each of its *transactive neighbors.*

Applicability: This process should be completed at the *update frequency* by *transactive nodes.*

Sub-Functions and Sub-processes:

4.1 Interpolate Intervals Service Functions—function, or set of functions, by which the inputs to this process may be restated using the current *interval start time (IST)* series. If input time series are found to use dated time intervals or any other representation of future intervals other than the current *IST* series, this function should be called until the dissimilarities are resolved. This function should also be called early during an *update interval* iteration to create updated, default versions of a recent prior *transactive feedback* *signals (TFS)* that may be used if, for any reason, this *transactive node* fails to formulate a *TFS* by the time it is used.

4.2 Predict Net Resource Surplus or Shortage—take the difference between total resource from *A* resources and total load supplied by this transactive node to determine the net surplus or shortage for each future interval *n*. The net surplus or shortage is the average power over an interval that should be sent to or received from transactive neighbors during that interval—an imbalance anticipated to occur at this *transactive node*. Therefore, the net surplus or shortage should equal the sum of all changes to the *TFS* for each interval at this *transactive node*.

$$\sum TFS_n = \sum_{a=1}^{A} \hat{P}_{G,a,n} - \sum \hat{L}_n \qquad \text{(Sub-Function } \mathbf{4.2}\text{)}$$

Total average load at each interval $\sum \hat{L}_n$ is a calculated input that should be retrievable from the **Predicted Inelastic and Elastic Load Buffer**. The total resource $\sum_{a=1}^{A} \hat{P}_{G,a,n}$ is a calculation available from the **Total Predicted Resource Buffer**, a product of **10. Sum Total Predicted Resource**. (Desirably, there is a connection between this calculated imbalance and resource planning.)

4.3 Disaggregate Net Resource Surplus or Shortage—allocate the net resource surplus or shortage among this *transactive node's transactive neighbors* by formulating or modifying the *TFS* for each such interface. The newly formatted *TFS* are then stored into the **Output TFS Buffer**.

Today, this prediction would rely on centralized power-flow solvers. In a fully distributed system, however, new prediction tools can be used.

This *transactive node* object should supply to this sub-function the current list of *transactive neighbors* for which *TFS* should be calculated. It may also provide simple ratios or detailed topological information that can be used eventually to predict load flow between this *transactive node* and its *transactive neighbors*, e.g., *TFS* series. Current information about the *transactive node* object is assumed to be available from a **Node State and Status Buffer**.

4.4 Refresh Default *Output TFS*—early during each *IST update interval*, this process should refresh the last calculated versions of *TFS* found in the **Output TFS Buffer** and restate them using the current *IST* series. Thereafter, the restated, refreshed *TFS* may be returned to the buffer and used as default values if, for any reason, this *transactive node* should fail to formulate the current *TFS* by the time they are used.

Inputs:

- Predicted total load supplied $\sum \hat{L}_n$ at each future interval *n* of the current *IST* series from the **Predicted *Inelastic* and *Elastic Load* Buffer**

- Predicted total resource $\sum_{a=1}^{A} \hat{P}_{G,a,n}$ at each future interval *n* of the current *IST* series. (This is now calculated by a sub-function of this process, but it can be made available from a common buffer of the *toolkit framework*.) This input should be available from the **Total Predicted Resource Buffer**.

- Information from this *transactive node* object concerning its *transactive neighbors* that should expect to receive a *TFS* from this *transactive node*, available from the **Node State and Status Buffer**.

- Information from this *transactive node's* object that will be used to allocate, or disaggregate, the net surplus or shortage among the *TFS* that are to be stated form each *transactive neighbor*, available from the **Node State and Status Buffer**.

- The current *IST* series available from the **Current *IST* Series Buffer**.

Outputs:

- One *output TFS* for each *transactive neighbor* stored into and available from the ***Output TFS* Buffer**.

Function / Process: Refer to the descriptions of the sub-functions above as the sub-functions were being introduced.

Dependencies:

- This process formulates one of two *transactive signal* types that should be available from the **Output TFS Buffer** to be conveyed by this *transactive node* to its *transactive neighbors* at the *update frequency* by **7. Send Transactive Signals.**
- This process expects that the current *IST* series will have been created by **2. Calculate New Transactive Signal Intervals** and available from the **Current IST Series Buffer.**
- This process expects that the current sum total load will have been calculated by function **5. Sum Total Predicted Load** and available from the **Predicted Inelastic and Elastic Load Buffer.**
- This process expects that the total predicted resource will have been calculated by function **10. Sum total Predicted Resource.**

Notes:

- The *TFS* is indeed a feedback signal, but the *transactive control and coordination system* is not a closed-loop feedback control system in the classical sense. First, the magnitude of resource from *responsive asset systems* is too small for us to expect closed-loop control. Second, the *TIS* is decidedly grounded as a meaningful delivered cost of energy, not free to represent large incentive swings as could a local marginal price. There is weak or no integral control in the system.
- The *transactive feedback signal (TFS)* may not be as dynamic and useful as the *transactive incentive signal (TIS)* will be. The TFS will be affected by a relatively small fraction of *responsive asset systems* at places throughout the *transactive control and coordination system.* Transmission and generation entities are unengaged by the project's scale and are therefore unresponsive to changes that will be observed in the *TFS.*

Figure **39** is a flowchart **3900** of an exemplary formulate TFS process

5. Sum Total Predicted Load

Purpose: Process to add the total inelastic (non-transactive) and elastic (transactive) electrical load components being supplied within the boundaries of this *transactive node*. (In the illustrated embodiment, electrical energy that is to be exported outside the boundaries of a *transactive node* is not part of this sum.)

Applicability: This function applies to *transactive nodes* and should be updated at the *update frequency*; however, this process becomes trivial for *transactive nodes* that supply no elastic electric load, no inelastic electric load, or neither elastic nor inelastic electric load within the boundaries of the *transactive node*.

Sub-Functions and Sub-processes:

5.1 Interpolate Intervals Service Functions—suite of functions that may be called upon should any inputs to this function note yet exist using the current set of *interval start times* that should be available from the **Current *IST* Series Buffer**.

5.2 Sum Inelastic Load—sums the entries in the **Inelastic Load Prediction Buffer** that are relevant to the current *update interval* iteration.

The **Inelastic Load Prediction Buffer** may (or may not) have a multiplicity of relevant entries that should be summed. For example the buffer might possess a bulk load prediction that is simply based on historical trends over the past week, the inelastic prediction for a large water heater responsive asset system, and the inelastic prediction for a voltage-response asset. (In certain embodiments, care should be taken not to double count any of the load as this sum is taken.) For each of this component addends *k*, the buffer should possess a relatively current entry $L_{inelastic,k}$. Each entry should state average load (unit: average power) to be consumed (or generated) by it during each of a series of intervals.

If an entry from the buffer is found to have intervals other than those in the current *IST* series, function **5.1 Interpolate Interval Service Functions** should be called upon to resolve the discrepancy and restate the entry contents using the current *IST* interval set.

Ideally, all current, relevant contents of the buffer will be evident from the entries' *interval start time* IST_0 time. Preferably, the buffer contents that are to found and summed by this sub-function for each iteration should be attributes of this *transactive node*, knowable from the contents of the **Node State and Status Buffer**.

The output product of this sub-function is a single time series $\sum L_{inelastic,n}$ that has summed components *k*.

5.3 Sum Change in Elastic Load—sums the entries in the **Toolkit Response Function Output Buffer** that are relevant to the current *update interval* iteration. If toolkit functions have been employed for responsive asset systems at this transactive node, one or more entries will be found in the buffer to be summed in this sub-function. Note that only the change in *elastic load* is to be found in the buffer and summed for each *interval start time* interval by this sub-function. For each of this component addends *j*, the buffer should possess a relatively current entry $\Delta L_{elastic,j}$. Each entry should state the change in average load (unit: average power) it predicted to be consumed (or generated) by it during each of a series of intervals.

If an entry from the buffer is found to have intervals other than those in the current *IST* series, function **5.1 Interpolate Interval Service Functions** should be called upon to resolve the discrepancy and restate the entry contents using the current *IST* interval set.

As was the case for sub-function **5.3** above, the contents of the buffer that are to found and summed by this sub-function for each iteration should be an attribute of this *transactive node*, knowable from the contents of the **Node State and Status Buffer**.

The output product from this sub-function is a single time series $\sum \Delta L_{elestic,n}$ that has summed components *j*.

5.4 Sum Total Inelastic and Change in Elastic Load—function by which total *inelastic load* predictions and predicted changes in *elastic load* are finally summed to calculate a total to be placed into the **Predicted Total Inelastic and Elastic Load Buffer**. This function completes the simple arithmetic sum

$$\sum L_{total,n} = \sum L_{inelastic,n} + \sum \Delta L_{elestic,n} , \qquad \text{(Function 5.)}$$

where $\sum L_{total,n}$ is the sum of total *inelastic load* $\sum L_{inelastic,n}$ and total change in *elastic load* $\sum \Delta L_{elastic,n}$ for *IST* interval *n* at this *transactive node*.

5.5 Refresh Predicted Total Inelastic and Elastic Load—succeeding calculations will expect that the predicted total *inelastic* and *elastic load* will be available according to current *IST intervals*. Therefore, early in each *update interval* interation, the most current representation of that sum should be located within the Predicted **Total *Inelastic* and *Elastic Load* Buffer and** subjected to function **5.1 Interpolate Intervals Service Functions** to recast the buffer contents into a default buffer entry that uses the current set of *interval start times (IST).* If for any reason this *transactive node* fails to later update its prediction of the sum into the buffer, the default value may be used instead.

Inputs:

- Set of predicted *inelastic load* $\{L_{inelastic,1}, L_{inealstic,2}, \ldots, L_{inelastic,k}, \ldots, L_{inelastic,K}\}$ for each of the K components of total *inelastic load*, each of which predicts average load (units: average power) for *interval start time* interval n. This set of entries should be found from within the **Inelastic Load Prediction Buffer.**

- Set of predicted changes to elastic load $\{\Delta L_{elastic,1}, \Delta L_{ealstic,2}, \ldots, \Delta L_{elastic,j}, \ldots, \Delta L_{elastic,J}\}$ for each of the J components of total change in *elastic load*, each of which predicts change in average load (units: average power) for *interval start time* interval n. This set of entries should be found from within the **Toolkit Response Function Output Buffer.**

- Current *interval start time (IST)* series from the **Current *IST* Series Buffer**

- List of those members of the **Inelastic Load Prediction Buffer**, if any, which are expected to be found and used by this process, which list should be obtained from attributes of this *transactive node* found in the **Node State and Status Buffer**.

- List of those members of the **Toolkit Response Function Output Buffer**, if any, which are expected to be found and used by this process, which list should be obtained from attributes of this *transactive node* found in the **Node State and Status Buffer.**

Outputs:

- Total predicted load $L_{total,n}$ for each of the current *IST intervals* to be stored into the **Predicted Inelastic and Elastic Load Buffer**.

Function / Process: The steps of this process were stated above with the introductions of sub-functions. Overall, the process completes the simple arithmetic sum

$$\sum L_{total,n} = \sum L_{inelastic,n} + \sum \Delta L_{elestic,n} , \qquad \text{(Function 5)}$$

where $\sum L_{total,n}$ is the sum of total *inelastic load* $\sum L_{inelastic,n}$ and total change in *elastic load* $\sum \Delta L_{elastic,n}$ for *IST* interval *n* at this *transactive node.*

Dependencies:

- Should call upon current *inelastic load* predictions from the **Inelastic Load Prediction Buffer** having been updated frequently by process **6. Predict Applicable Inelastic Load using Trends / Models**.

- For those transactive nodes that have *responsive asset systems* and therefore employ *toolkit functions*, this function expects that current predictions of changes in *elastic load* are available from the **Toolkit Response Function Output Parameter Buffer** having been updated frequently by process **Calculate Applicable Toolkit Response Function(s).**

- The output from this function is an input to process **4. Formulate *TFS*.**

Notes:

- If the prediction of current total *elastic* and *inelastic load* components cannot be calculated promptly by the time they are used by the *transactive node*, prior calculations from the **Predicted *Inelastic* and *Elastic Load* Buffer** should be used by process **4. Formulation *TFS*.**

- It would be ideal if inputs into and outputs from this function were properly formatted using the current *interval start time series (IST)* that should exist in the **Current *IST* Series Buffer**. Keeping the current outputs of functions and processes aligned with the current IST series will greatly simplify later successive calculations. If that cannot beS accomplished, interpolation service functions should be called upon.

- Implementers might choose to have this process additionally interact with the system management layer. If, for example, this transactive node fails to update its load predictions and therefore uses default, buffered estimates, such events might be counted and/or flagged to initiate notifications or alerts. Such a capability would be nice to have, but it is probably not an essential part of the *toolkit framework*. System management for this process would serve business entities that are relevant to the "generic" system implementation.

Figure **40** is a flowchart **4000** of an exemplary sum total predicted load process.

6. Calculate Applicable Toolkit Load Functions

Purpose: This process block represents from zero to many specific *toolkit library functions* that may be incorporated into the *toolkit framework* here. The *toolkit functions* that become instantiated at this location should represent and predict *elastic* and *inelastic loads* and should result in a reasonably complete and accurate prediction of the entire load that is supplied within the boundaries of this *transactive node* during each *IST* interval.

Most generally, these *toolkit functions* may be characterized by their inputs and outputs and by their generalized functional responsibilities within the *toolkit framework*. A template is developed for the specification of *toolkit functions* (see SubAppendix B). Owners of *transactive nodes*, who represent the unique perspective under which this *transactive node* should be managed, should select and/or help create specific *toolkit function(s)* that model the *responsive asset systems* and *inelastic loads* that they have or plan to implement. See Table **25** for an example list of *toolkit load functions*.

Modular *toolkit functions* may be implemented and shared via combinations of their functional descriptions, pseudo code implementations, and reference code, all of which are recommended components of the recommended *toolkit function* template.

The location of this block within the *toolkit framework* is intended for *toolkit functions* that predict the behaviors of two different types of loads:

- *responsive asset system*—an *elastic load m* for which its *toolkit function* predicts both its *inelastic load* component L_m and a change in *elastic load* ΔL_m using the current *output TIS* and often *other local conditions* as inputs.

- *inelastic load*—other *inelastic load* component for which its toolkit function predicts only its inelastic load component L_m.

Of interest are those *responsive asset systems* that can be applied to the *transactive control and coordination system*. (In certain embodiments, *responsive asset systems* have been defined to be applied within reliability or conservation and efficiency test cases as well. Not all *responsive asset systems* are being used in the *transactive control and coordination system* test cases.) A *toolkit function* should be defined for each unique implementation of each major type of *responsive asset system*. Each *toolkit function* should first calculate the inelastic load L_m, which predicts when and how much energy the *responsive asset system* would consume if it were not influenced by the *output TIS*. The prediction of *inelastic load* component is placed into the **Inelastic Load Prediction Buffer**. The *toolkit function* should then predict the *change* in *elastic load* ΔL_m that is caused by the condition of the *output TIS*. The prediction of *elastic load* component is placed into the **Elastic Load Prediction Buffer**. It is acceptable that the *elastic load* components may be zero during intervals when the *responsive asset system* is not predicted to be engaged by the *output TIS*.

Another output from a *toolkit function* should be a representation of the planned control action by which the *responsive asset system* will be induced to change its energy consumption in light of the state of the *output TIS* for each interval. For example, some *responsive asset systems* may be either active or curtailed (e.g., populations of water heaters), in which case a binary indicator might be used for each interval. Other systems are able to enter any of multiple discrete levels of response (e.g., GE smart appliances), in which case one of several discrete levels should be specified for each interval. Still other systems may provide a continuum of possible responses and use a representation of percentage. (An interesting example of this continuum of responses will occur where customers are provided a means to view the output TIS itself on an in-home display and respond correspondingly with a continuum of behavioral responses.) Eventually, as time marches toward the interval of interest and the interval becomes that of IST_0, the responsive asset system should be expected to take the predicted, prescribed action. The implementations of *responsive asset systems* will be diverse, but it is in the representation of these predicted, planned control actions where standardization may be particularly useful.

An example would probably be useful concerning the portion of predicted load that should be included in this process from *elastic loads*, including *responsive asset systems*. Electrical consumption by a set of electric water heaters may be predicted quite well from measured trends and models of the water heaters and their owners' behaviors. The input information or parameters that influence such trends and models might include time of day, day of week, occupancy, outdoor temperature, and average outdoor temperature, for examples. In the *toolkit framework*, these pieces of information or parameters are referred to as *other local conditions* that should be available inputs if the *transactive node* is to accurately predict the load consumed by the water heaters. These predictions are to be completed within this process **6. Predict Applicable *Toolkit Load Functions***. The predicted load should be recorded for each such system in the ***Inelastic Load* Prediction Buffer**. If upon receipt of the current *output TIS* the water heaters would reduce their load, the change (e.g., only the change) would be predicted in a parallel calculation path and would be stored into the ***Elastic Load* Prediction Buffer**.

Toolkit functions can used to describe behaviors of individual devices. But the *responsive asset systems* of the *Demonstration* are primarily used for populations of devices. It is the statistical behavior of the populations, not individual devices that should be predicted.

Inelastic load components are similarly incorporated via their *toolkit functions*; however, no *elastic load* component should be created by these functions. Candidate *inelastic load* predictions might include feeders of residential customers, where the load of the population could be predicted from the time of day, average home square footage, average house age, outdoor temperature, and perhaps still *other local conditions*.

Regardless of whether a given toolkit function describes an elastic or an inelastic load, a load should never appear on both the resource and load sides of the *toolkit framework* formulation for any single interval *n*. *Responsive asset systems* may be either electrical loads or resources. Regardless, the *toolkit functions* whose influence is to be inserted at this location will affect the formulation of the *TFS* but will not directly influence the formulation of the *output TIS*. *Responsive asset systems* that should affect the delivered cost of energy (e.g., the *TIS*) at this *transactive node* should be inserted at location **8. Calculate Applicable Toolkit Resource Functions** instead.

Using the above-stated criterion, the average power from a customer's renewable generator should probably be treated as a "negative" load (e.g., its *toolkit function* should be incorporated here) if it will never result in net metering. But if the utility at any time pays the customer net-metering payments for surplus energy that is produced by the resource, the resource should be included instead among resources, not loads, so that the net-metering charges may influence the formulation of the *TIS* (e.g., a *toolkit function* should be included for this system in the process **8. Calculate Applicable Toolkit Resource Functions**).

Using the same reasoning, the present process should not predict bulk generation resources that are scheduled at this *transactive node* because costs should almost certainly be applied to the energy from such bulk resources.

The influences of *elastic* and *inelastic load* components should never be double counted. The influence of a load should appear only once if an accurate prediction of total load is to be formulated by this *transactive node*.

Toolkit functions may include learning algorithms and other means to improve the accuracy of their load predictions over time, but such complexities should be weighed against the *Demonstration's* desire to create and teach and implement these *toolkit functions* with its participants and within a tight development schedule.

See Table **25** for a list of example *toolkit load functions*.

Applicability: Any *toolkit functions* to be called upon in this process block should be called at the *update frequency*. It is conceivable but unlikely that a *transactive node* may have neither *inelastic* nor *elastic load* components that necessitate any *toolkit functions* be called within this process block.

Sub-Functions and Sub-processes:

6.1 Interpolate Intervals Service Functions—a suite of service functions that may be called upon as they are desired to restate dated time series in terms of the current *IST* intervals. (These functions might be defined and used throughout the entire *toolkit framework* instead of uniquely defined for each process, as has been shown here.)

6.2m Toolkit Load Function—from zero to many individual *toolkit functions* from a *toolkit function library* that predict *inelastic load* and change in *elastic load* for each interval of the current *IST* series. Enough such *toolkit functions* should be incorporated and called upon to predict the entire load at this *transactive node*. Individual t*oolkit functions* may be created or selected from a *toolkit function library* predict the behaviors of a *responsive asset system*; the behaviors of a group of *inelastic loads*; generation from small distributed generation resources that do not directly influence the formulation of the *TIS*; or large nebulous groups of ill-defined loads that can only be characterized by their historical trends.

It should be assumed that the list of *M* relevant *toolkit functions* are identified and known by this *transactive node* object and is available from the **Node State and Status Buffer**. Furthermore, the buffer should identify the sets of *other local conditions* inputs expected to be available to the M *toolkit functions* from the **Toolkit Load Function Input Buffer**.

A *toolkit function* should output its prediction of *inelastic load* into the ***Inelastic Load*** **Prediction Buffer** for the load being described and for a current *IST* interval. (The inputs expected by toolkit functions will be varied and may be dynamic.) If the function models and helps control responsive, *elastic* loads, the function should also create and output the planned control for the responsive load. A standardized advisory control signal to be sent to the *responsive asset systems* has been formulated and is available in SubAppendix C.

6.3 Refresh Predicted Inelastic Elastic Loads—early each *update interval* iteration, the most current contents of the ***Inelastic Load*** **Prediction Buffer** and ***Elastic Load*** **Prediction Buffer** should be retrieved by this sub-function and restated using **6.1 Interpolate Intervals Service Functions** in terms of the current *IST* interval set. These updated buffer contents are then available to be used by default should this *transactive node* fail for any reason to calculate its load for the current iteration.

Inputs:

- current *IST* interval series that is available from the **Current *IST* Series Buffer**

- current *output TIS* (units of "value" attribute: cost per energy) from the **Output TIS Buffer**

- *other local conditions (OLC)* (units: various) as might be prescribed by specific *toolkit functions* and available from the **Toolkit Load Function Input Buffer**

- list of *M toolkit functions* that should be called at this *transactive node* and the list of *other local conditions* data inputs that will be used by the *toolkit functions* as should be known by this transactive node object and available from the **Node State and Status Buffer**.

Outputs:

- *Inelastic load* predictions L_m (units: average power) for each *IST* interval stored into the **Inelastic Load Prediction Buffer**

- *Elastic load* predictions ΔL_m (units: change in average power) for each *IST* interval stored into the **Elastic Load Prediction Buffer**

- Predicted control actions for *responsive asset systems* for each *IST* interval stored into the **Elastic Load Prediction Buffer**. (recommendation for units: {allowed: "**0**"; curtailed: "**-1**"}; {generation level L: "L"; ..., generation level **2**: "**2**"; generation level **1**: "**1**"; off: "**0**"; load reduction level -**1**: "-**1**"; load reduction level -**2**: "-**2**"; ...}; {continuum from full generation: "**100**"; off: "**0**"; full load reduction: "-**100**})

Function / Process: Sub-functions **6.1** and **6.3** were described as they were being introduced in the text above. This document has stated functional responsibilities and an input/output model for the multiplicity of *toolkit functions* **6.2m Toolkit Load Function** that are to be called upon during this process. Each *toolkit function* should use the provided template and should describe for itself what it is meant to accomplish within the functional responsibilities, inputs, and outputs that have been generally described here.

Dependencies:

- The current *TIS* should have been calculated by **3. Formulate *TIS*** and available from the ***Output TIS* Buffer**.

- Various current *other local conditions* should be available from the **Toolkit Load Function Input Buffer**. The list of relevant *other local conditions* should be known to the *transactive node* object and available from the **Node State and Status Buffer**. Note that the *other local conditions* might themselves use management of other data collection and maintenance systems and processes.

- A list of functions **6.2m Toolkit Load Functions** should be unambiguously named and known to this *transactive node* object available from the **Node State and Status Buffer**.

- Process **2. Calculate New Transactive Signal Intervals** should have run recently to provide to this process current *IST* intervals available from the **Current *IST* Series Buffer**.

- This process inserts up to *M* entries into each the ***Inelastic Load* Prediction Buffer** and ***Elastic Load* Prediction Buffer**, one for each *toolkit function* that is called. The contents of these buffers should be current and available to be summed by process **5. Sum Total Predicted Load.**

Notes:

- No load or resource should appear on both the load and resource sides of the toolkit formulation for any given *IST* interval.

- The sum of the *inelastic load* stored into the ***Inelastic Load* Prediction Buffer** and change in *elastic load* stored into the **Toolkit Response Function Output Buffer** for a give *toolkit function* should closely predict the actual load, providing the *TIS* and *other local* *conditions (OLC)* remain about the same until the corresponding *IST_n* interval becomes *IST_0*.

- It is hoped but not required that model-based predictions of both the *inelastic-load* and change in *elastic-load* components may improve over time as more sophisticated *toolkit functions* use historical feedback to improve their algorithms.

- Implementers are encouraged to use this process and its toolkit functions for model-based load predictions, regardless of whether they describe *elastic load*.

Table 25: Example Resource, Incentive and Load *Toolkit Functions*

Resource or Incentive	Load
1.0 Imported Electrical Energy **1.1** Non-Transactive Imported Energy **1.2** Transactive Imported Energy **2.0** Renewable Energy Resource **2.1** Wind Energy **2.2** Solar Energy **2.3** Hydropower **3.0** Fossil Generation **4.0** General Infrastructure Cost **5.0** System Constraints **5.1** Transmission Flowgate **5.2** Equipment and Line Constraints **6.0** System Energy Losses	**1.0** Bulk Inelastic Load **1.1** Bulk Commercial Load **1.2** Bulk Industrial Load **1.3** Bulk Residential Load **1.4** Small Wind Generator Negative Load **1.5** Small-Scale Distributed Generator Negative Load **1.6** Small-Scale Solar Generator Negative Load **2.0** General Event-Driven Demand Response **2.1** Commercial Event-Driven Demand Response **2.2** Event Driven Distribution System Voltage Control **2.4** Residential Event-Driven Demand Response **2.5** Non-Renewable Distributed Generation Event-Driven Demand Response **3.0** General Time-of-Use Demand Response **3.1** Battery Storage--Time-of-Use **3.2** Commercial Time-of-Use Demand Response

6.**1** Transmission Losses	**3**.**4** Residential Time-of-Use Demand Response
6.**2**. Distribution Losses	**3**.**5** Time-of-Use Distribution System Voltage Control
7.**0** Demand Charges	**3**.**6** Time-of-Use Electric Vehicle Charging
7.**1** BPA Demand Charges	**4**.**0** General Real-Time Continuum Demand Response
8.**0** Market Impacts	**4**.**1** Battery Storage--Real-Time
8.**1** Spot Market Impacts	**4**.**2** Commercial Real-Time Demand Response
	4.**3** Real-Time Distribution System Voltage Control
	4.**5** Residential Real-Time Demand Response
	5.**0** General Manual or Behavioral Demand Response
	5.**1** Residential Behavioral Response to Portals or In-Home Displays
	5.**2** Residential Behavioral Response--No Portals or In-Home Display
	5.**3** Manual Commercial Demand Response
	5.**4** Manual Non-Renewable Distributed Energy Resources Demand Response

Figure **40** is a flowchart **4000** of an exemplary “calculate applicable toolkit load functions” process.

7. Send *Transactive Signals*

Purpose: Method by which *output transactive signals* are conveyed from this *transactive node* to each one of its *transactive neighbors.* Most generally, there will be no single approach to completing this process because transactive is tied to no single communication technology, medium, or protocol. *Transactive neighbor* pairs should negotiate and agree upon these details. On the other hand, the *Demonstration* has elected to convey *transactive signals* almost exclusively via secure Internet.

Applicability: An process that should be completed at the *update frequency* by a *transactive node*.

Sub-Functions and Sub-processes: The following high-level responsibilities should be addressed, regardless of the platforms on which it is designed:

- Format *transactive signals* according to published recommendations, including published XML schema.
- Coordinate timing with *transactive node* object states during each *update interval* and iteration
- Compare and coordinate *Transactive neighbor* list with *transactive node* object state.

Inputs:

- One *output TIS* series from process **3. Formulate TIS**
- One *output TFS* series for each *transactive neighbor* from process **4. Formulate TFS**

Outputs:

- Paired couples of *output TIS* and *output TFS* sent to each *transactive neighbor.*

Dependencies:

- Receives current *output TIS* from **Output TIS Buffer**

Notes:

- This process or function is trivial from a functional perspective, but it is useful from a system interoperability perspective. *Transactive nodes* that employ unlike software and computational architectures should still be able to send and receive these signals from their *transactive neighbors.*

This function or process is also useful from a cyber-security perspective. Both the senders and recipients of transactive signals should be satisfied that their systems will remain safe from attack.

Figure **43** is a flowchart **4300** of an exemplary "send *transactive signals*" process.

8. Calculate Applicable Toolkit Resource and Incentive Functions

Purpose: A multiplicity of *toolkit functions* may be applied at this location within the *toolkit framework* to address resources and incentives. *Toolkit functions* should be created or selected from a *toolkit library* to represent the energy resources and incentives that are be applied at this *transactive node* during each *IST* interval. The costs that are calculated by the *toolkit functions* in turn may incentivize or disincentivize consumption and generation of electricity through their effects on the *transactive incentive signal.*

See Table **25** for a list of example *toolkit resource and incentive functions.* Refer to SubAppendix B for a template that may be used to specify additional *toolkit resource and incentive functions* as they are developed.

Applicability: A *transactive node* should calculate at least one *toolkit function* at the *update frequency.*

Sub-Functions and Sub-processes:

8.1 Interpolate Intervals Service Functions—a suite of service functions that can accept stale, dated data and restate the data in terms of the current *IST* interval series. (These functions might be defined and used throughout the entire *toolkit framework* instead of uniquely defined for each process, as has been shown here.)

8.2 Refresh Predicted Resources and Incentives—Early during each *update interval*, this sub-function retrieves the most recent entries from the **Resource Schedules and Cost Buffer** and restates the records in terms of the current *IST* series. If for any reason this transactive node fails to complete the present process by the time its outputs are used, the restated records may be used as default records.

8.3 Assign Energy Cost and Average Power—a sub-function of a *toolkit resource and incentive function* in which cost $C_{E,a,n}$ (units: cost per energy) is assigned to each component a of energy $\hat{P}_{G,a,n}$ (units: average power) that is either imported into or generated within the boundaries of this *transactive node.* In particular embodiments, one responsibility of a *toolkit resource and incentive function* is to calculate and report one of each of these two quantities for each current *IST* interval *n.* Either of the calculated quantities may be zero. The calculated values will differ depending on selected *toolkit function* and the resource or effect that is being modeled by the selected *toolkit function.*

Example energy costs and energies that that should be captured using this sub-function include

- The cost of energy from traditional bulk generation

- The cost of energy from renewable energy resources like wind. (Wind energy is desirably incentivized by applying its costs to its infrastructure and not to the energy that is produced. Thereby, it causes a downward influence on the delivered cost of energy at the time and near where wind blows.)

- For non-transactive neighbors, the cost of energy that applies to any energy that is imported into the boundary of this *transactive node*. (Note that if energy is exported rather than imported during an *IST* interval *n*, it is not counted among resources, so either or both the energy terms for this sub-function should be set to zero.)

- For transactive neighbors, the cost of delivered energy (e.g., the *TIS*) that applies to imported energy (e.g., the *TFS*). (This is a special case where the *input TIS* and *input TFS* are read from the ***Input Transactive Signal*** **Buffer**. A simple *toolkit function* should be created to complete this task.)

The values $C_{E,a,n}$ should be defensible representations of the delivered costs of energy $\hat{P}_{G,a,n}$.

The sum of $\hat{P}_{G,a,n}$ should represent the energy that is generated within or imported into this *transactive node* during *IST* interval *n*.

This sub-function may call upon various defined *other local conditions* that should be available as inputs from the *Resource and incentive Input Buffer*. The list of *other local conditions* that are expected by a give *toolkit function* should be known by the *transactive node* object and available from the **Node State and Status Buffer**.

Refer to sub-function **3.7 Calculate *Output TIS*** to fully understand how the two outputs from the present sub-function will become incorporated into the formulation of *TIS* within the *toolkit framework*.

8.4 Assign Capacity Cost and Capacity— a sub-function of a *toolkit resource and incentive function* in which cost $C_{C,b,n}$ (units: cost per power) is assigned to capacity limitations and costs that are triggered by capacities. The sub-function also captures the capacity $\hat{P}_{C,b,n}$ (units: average power) to which the cost applies. In certain embodiments, one responsibility of a *toolkit resource and incentive function* is to calculate one of each of these two quantities for each current *IST* interval *n*. Either of the calculated quantities may be zero. The calculated values will differ depending on selected *toolkit function* and the resource or effect that is being modeled by the selected *toolkit function.*

Example capacity costs that should be included through this sub-function include

- Costs that should be applied as equipment like power lines become constrained
- Imposed demand charges that become applied to the owners of this *transactive node.*

Cost $C_{C,b,n}$ should be defensible as cost that will be incurred upon a corresponding capacity $\hat{P}_{C,b,n}$ that is predicted to occur during *IST* interval *n*.

This sub-function may call upon various defined *other local conditions* that should be available as inputs from the *Resource and incentive Input Buffer.* The list of *other local conditions* that are expected by a give *toolkit function* should be known by the *transactive node* object and available from the **Node State and Status Buffer**.

Refer to sub-function **3.7 Calculate *Output TIS*** to fully understand how the two outputs from the present sub-function will become incorporated into the formulation of *TIS* within the *toolkit framework.*

8.5 Assign Infrastructure Cost— a sub-function of a *toolkit resource and incentive function* in which cost $C_{I,c,n}$ (units: cost per time) is assigned to the provision of infrastructure at this *transactive node*, which costs are usually spread over quite long periods of time. In certain embodiments, one responsibility of *toolkit resource and incentive function* is to calculate and report one infrastructure cost output for each current *IST* interval *n*. Its value may be zero. The calculated value will differ depending on selected *toolkit function* and the resource or effect that is being modeled by the selected *toolkit function.*

Example infrastructure costs that may be used through this sub-function include

- Initial purchase costs for equipment

- Initial installation costs
- Maintenance costs.

Refer to sub-function **3.7 Calculate *Output TIS*** to fully understand how the output from the present sub-function will become incorporated into the formulation of *TIS* within the *toolkit framework.*

8.6 Assign Other Costs— a sub-function of a *toolkit resource and incentive function* in which other costs (units: cost) that cannot be represented by the other sub-functions are applied at this *transactive node*. In certain embodiments, one responsibility of a *toolkit resource and incentive function* is to calculate and report one such other cost output for each current *IST* interval *n*. Its value may be zero. The calculated value will differ depending on selected *toolkit function* and the resource or effect that is being modeled by the selected *toolkit function.*

This sub-function should not be used to bypass the other three sub-functions **8.3**, **8.4**, and **8.5**. The other cost that is assigned by this sub-function should be a defensible component of the delivered cost of energy (e.g., the *TIS*) that will be formulated by process **3. Formulate *TIS*.**

Refer to sub-function **3.7 Calculate *Output TIS*** to fully understand how the output from the present sub-function will become incorporated into the formulation of *TIS* within the *toolkit framework.*

Inputs:

- Current *Input TIS* and *TFS* should have been received in process **1. Receive *Transactive Signals*** and should be available from the **Input Transactive Signal Buffer**. These inputs will be treated the same as other energy terms.
- Current *other local conditions* data that has been specified for by the set of *toolkit functions* that are being applied at this *transactive node.*
- The list of *toolkit functions* that are to be applied in this process, which list should be known to this *transactive node* object and available from the **Node State and Status Buffer.**

- The list of *other local conditions* data records that are expected by the set of *toolkit functions* that are employed in this process block, which list should be known by this *transactive node* object and available from the **Node State and Status Buffer**.

Outputs:

- One paired energy cost and energy ($C_{E,a}$. $\hat{P}_{G,a}$) series record placed into and available from the **Resource Schedules and Cost Buffer** for each of the *toolkit functions* that is applied within this process. (There are *A* non-zero of these records used to represent imported and generated energy.)

- One paired capacity cost and capacity ($C_{C,b}$. $\hat{P}_{C,b}$) series record placed into and available from the **Resource Schedules and Cost Buffer** for each of the *toolkit functions* that is applied within this process. (There are *B* non-zero of these records where capacity costs are relevant.)

- One infrastructure cost $C_{I,c}$ series record placed into and available from the **Resource Schedules and Cost Buffer** for each of the *toolkit functions* that is applied within this process. (There are *C* non-zero of these records where infrastructure costs are relevant.)

- One other cost $C_{O,d}$ series record placed into and available from the **Resource Schedules and Cost Buffer** for each of the *toolkit functions* that is applied within this process. (There are *D* non-zero of these records where other costs are relevant.)

Function / Process: The sub-functions were described above as they were being introduced. Sub-functions **8.3**, **8.4**, **8.5**, and **8.6** are components of *toolkit functions* and may not be generically defined except through the characterization of their inputs and outputs.

Dependencies:

- This process should find current current input *transactive signals* from process **1. Receive *Transactive Signals*** from within the ***Input Transactive Signal* Buffer**.

- This process expects current and relevant *other local conditions* are available from the **Resource and Incentive Input Buffer**. The list of example *other local conditions* records is known to this *transactive node* object and available from the **Node State and Status Buffer**.

- This process expects that the relevant list of *toolkit functions* will be known to the *transactive node* object and available from the **Node State and Status Buffer**. The modular *toolkit functions* themselves should be available at the *transactive node*.

- This process expects that the current *IST* series will have been calculated by process **2. Calculate New *Transactive Signal* Intervals** and will be available from the **Current *IST* Series Buffer**.

- This process outputs to the **Resource Schedules and Cost Buffer** that are used for processes **10. Sum Total Predicted Resource** and by **3. Formulate TIS**.

Notes:

- A transactive node should instantiate at least one *toolkit function* that redefines current *transactive signals* as energy terms and places them into the **Resource Schedules and Cost Buffer**.

- General guidance should be that a *transactive control and coordination system* can address economic decisions that interact with the system somewhat slower than the *update frequency*. There will occur an interim period where the *Demonstration's* system will accept but not influence resource decisions that presently involve markets and ancillary services that are not initially tied into the *transactive control and coordination system*. However, many such economic decisions may be addressed and perhaps optimized by a *transactive control and coordination system* as theories are developed to support doing so.

- An alternative pathway has been provided for “Scheduled Resources” to become entered into the **Resource Schedules and Cost Buffer**. It is preferred, however, that even non-transactive resources enter into the *toolkit framework* via a toolkit function and this process **8. Calculate Applicable *Toolkit Resource and Incentive Functions***. One of our most basis toolkit functions should be one that represents traditional, bulk generation.

Figure **43** is a flowchart **5300** of an exemplary “calculate applicable *toolkit resource and incentive functions*” process.

9. Control Responsive Asset Systems

Purpose: Advise *responsive asset systems* of the actions that they should take during the present *update interval* in accordance with their planned responses for the current *interval start time* IST_0.

Applicability: This process should be completed at the *update frequency* by a *transactive node* that has at least one *responsive asset system* installed and responsive to the *transactive control and coordination system.*

Some *transactive node* owners will impose constraints on the dynamics with which their *responsive asset systems* may act, in which case this process may be completed less frequently than the *update frequency*. For example, certain *responsive asset systems* may be engaged only at the top of an hour and may remain engaged for minimum durations after that. Still others should be scheduled some time prior and are therefore not responsive to the *update frequency*. (The capabilities of various responsive asset systems are desirably addressed in the selected toolkit library functions **6.2m Toolkit Load Function.)**

Sub-Functions and Sub-Processes: None. This process may be only described at a functional level due to the diversity of the *responsive asset system* that is to be controlled. Most of the actual control activities take place within the *responsive asset systems* themselves and according to the preferred practices of this *transactive node's* owner.

Inputs:

- Advisory signal for *current interval start time* IST_0 that is available from the ***Elastic Load* Prediction Buffer**. Each of these inputs is expected to have one of three meanings depending upon the capabilities of the targeted responsive asset system—discrete binary, discrete multilevel, or continuous.

Outputs:

- The principal output actually occurs outside the *transactive control and coordination system* and outside this process but in the final control of the assets within the target *responsive asset system.*

- The state or status of the responsive asset system may be updated to the **Node State and Status Buffer**. For example, this buffer may hold information about the availability of the system or the amount of load that is presently available to be controlled.

Function / Process: The process by which the advisory output found within the ***Elastic Load* Prediction Buffer** is to be converted into control actions for the present *update interval* will be quite unique to the *responsive asset system* and will take place within the system according to practices of this *transactive node's* owner.

Dependencies: If this transactive node possesses any *responsive asset systems*, then

- This process expects to find a current advisory response for each respective *responsive asset system* having been predicted (planned) by its respective process **6. Calculate Applicable *Toolkit Load Functions*** and available in the ***Elastic Load* Prediction Buffer**. Only the current *interval start time* IST_0 is relevant to the actual, not the planned, control of a *responsive asset system.*

Notes:

- Note that the *toolkit function* that corresponds to a given *responsive asset system* should state the information about the system that should be maintained within the **Node State and Status Buffer.**

- *Responsive asset systems* may be either energy loads or resources.

- The *transactive control and coordination system* advises a *responsive asset system* via this process, but it never directly controls any *responsive asset system.* A *responsive asset system* is not part of the *transactive control and coordination system.*

- Responsive asset systems are very diverse. Even similar asset systems use different approaches, practices, protocols, and standards. One might realize an opportunity for standardization in the three types of signals that will be used to advise control actions for *responsive asset systems*—discrete binary, discrete multilevel, and continuous.

- For the *Demonstration*, *responsive asset systems* almost exclusively refer to populations of individual assets. The *Demonstration's transactive control and coordination system* therefore will provide an advisory "control" signal to the system, not to its individual assets. If use of *transactive control and coordination systems* continues, it is feasible that they will be extended down to individual assets. In principle, a *transactive control and coordination system* is very scalable.

Figure **44** is a flowchart **5400** of an exemplary "control *responsive asset systems*" process.

10. Sum Total Predicted Resources

Purpose: Sum the total energy resources entering the boundaries of this *transactive node*. The transactive node that has A resources

The sum produced by this process is used for two purposes in the *toolkit framework*: First, it is the divisor in process **3. Formulate *TIS***. Second, during process **4. Formulate *TFS*** it is compared against the total load that is calculated by process **5. Sum Total Predicted Load**, resulting in the net surplus or shortage of energy that should be allocated among the *TFS* of of *transactive neighbors*.

Applicability: This process should be completed at the *update frequency* by a *transactive node*.

Sub-Functions and Sub-processes:

10.1 Interpolate Intervals Service Functions— a suite of service functions that may be called upon as they are desired to restate dated time series in terms of the current *IST* intervals. (These functions might be defined and used throughout the entire *toolkit framework* instead of uniquely defined for each process, as has been shown here.)

10.2 Sum Total Predicted Resource—sum of the *A* resources $\hat{P}_{G,a,n}$ (units: average power) for each *IST* interval *n*. This sub-function should find a current representation of each summand from within the **Resource Schedules and Cost Buffer**. The expected set of summands should be known to this *transactive node* object and available from the **Node State and Status Buffer**. The sum should include electrical energy that is either generated within or imported into the boundaries of this *transactive node* during each *IST* interval *n*. Each of the summands should be found paired with an energy cost parameter C_E in the **Resource Schedules and Cost Buffer**.

Summands $\hat{P}_{G,a,n}$ should include and represent

- The *TFS* (units: average power) of each *transactive neighbor* from which this *transactive node* will import energy during interval *n*.

- The average energy generated during *IST* intervals *n* from any generator within the boundaries of this transactive node which may be expected to influence the formulation of the *TIS*. That is, its generated energy should be paid for and represented in the *transactive control and coordination system*. (This will include almost all generation resources. An exception will be generation by end-use customers that displaces their load but never should affect the cost energy in a way that would be evident outside the customer premises.)

- Energy imported during *IST* intervals *n* from electrically connected neighbors who are not *transactive neighbors*.

$$\text{Total Predicted Resource} = \sum_{a=1}^{A} \hat{P}_{G,a,n} \qquad \text{Process } \mathbf{10}$$

The output product from this sub-function is a single time series (units: average power) placed into the **Total Predicted Resource Buffer** each *update interval*.

10.3 Refresh Predicted Total Resource— early each *update interval* iteration, the most current contents of the **Total Predicted Resource Buffer** should be retrieved by this sub-function and restated using **10.1 Interpolate Intervals Service Functions** in terms of the current *IST* interval set. These updated buffer contents are then available to be used by default should this *transactive node* fail for any reason to calculate total resource for the current iteration.

Inputs:

- A multiplicity of resource components $\hat{P}_{G,a,n}$ (units: average energy) to be retrieved from the **Resource Schedules and Cost Buffer**.
- The identifiers of *A* resource components known by this *transactive node* object and available from the **Node State and Status Buffer**.
- Current *interval start time (IST)* series available from the **Current *IST* Series Buffer**.

Outputs:

- Sum of resources $\sum_{a=1}^{A} \hat{P}_{G,a,n}$ (units: average power) stored into the **Total Predicted Resource Buffer**. This output is a series of values, one for each *IST* interval.

Function / Process: The purpose of this process is to perform a mathematical sum, which has been described above as the sub-functions were being introduced.

Dependencies:

- This process uses a current *IST* series to have been calculated by process **2. Calculate New *Transactive Signal* Intervals** and available from the **Current *IST* Series Buffer**.

- This process expects that current resource components $\hat{P}_{G,a,n}$ will have been placed into the **Resource Schedules and Cost Buffer** by process **8. Calculate Applicable *Toolkit Resource and Incentive Functions***. However, the sub-**function 8.3 Refresh Predicted Resources and Incentives** will have created a default set of inputs that may be used here if current inputs cannot be calculated.

- The current output of this process is used by process **3. Formulate *TIS*** and **4. Formulate *TFS*** and is expected to be available from the **Total Predicted Resource Buffer**. However, some resiliency is provided by sub-function **10.3 Refresh Predicted Total Resource**, which calculates a default current process output to be available from the **Total Predicted Resource Buffer** should this process fail to create a current output by the time it is used.

Notes:

- Refer to processes **3. Formulate *TIS*** and **4. Formulate *TFS*** that will give one a better sense of how the output of this process is to be used.

- The general term $\hat{P}_{G,a,n}$ has been introduced, in part, to deemphasize that there are multiple types of such terms, including even the *TFS* at time it describes imported energy. Altogether, these terms should include the energy that is generated or imported within this *transactive node's* boundary.

This process was originally considered as a sub-function within both processes **3** and **4**. Because both processes performed the identical function, the function was elevated to a process at the *toolkit-framework* level so that the same sum may be used by both processes **3** and **4**.

Figure **45** is a flowchart **4500** of an exemplary "sum total predicted resources" process.

11. Control Responsive Resource

Purpose: Advise *responsive resources* of the actions that they should take during the present *update interval* in accordance with their planned responses for the current *interval start time* IST_0.

Applicability: This process should be completed at the *update frequency* by a *transactive node* that has at least one *responsive resource*. This process will be used infrequently until resources like bulk generators become responsive to a dynamic *transactive control and coordination system*.

Some *resource* owners will impose constraints on the dynamics with which their *resources* may act, in which case this process may be completed less frequently than the *update frequency*.

Sub-Functions and Sub-Processes: None. This process may be only described at a functional level due to the diversity of the resources that are to be controlled. Most of the responsibilities to engage resources lie with the resource systems themselves and not with processes of the *toolkit framework*.

Inputs:

- Resource plans as formulated by certain *toolkit functions* within the process **8. Calculate Applicable Toolkit Resource and Incentive Functions.**

Outputs:

- The principal output actually occurs outside the *resource system* and outside this process but in the final control of the resource within the target resource system.

- The state or status of the resource may be updated to the **Node State and Status Buffer**. For example, this buffer may hold information about the availability of the system or the amount of resource that is presently available to be controlled.

Function / Process: The process by which the advisory output found within the **Resource Schedules and Cost Buffer** is to be converted into control actions for the present *update interval* will be quite unique to the responsive resource system and will take place within the system according to practices of the resource and *transactive node* owners.

Dependencies: If this *transactive node* possesses any responsive resource systems, then

- This process expects to find a current advisory response for each respective responsive resource system having been predicted (planned) by its respective process **8. Calculate Applicable Toolkit Resource and Incentive Functions** and available in the **Resource Schedules and Cost Buffer**. Only the current *interval start time* IST_0 is relevant to the actual, not the planned, control of a responsive resource system.

Notes:

- Note that the *toolkit function* that corresponds to a given responsive resource system should state the information about the system that should be maintained within the **Node State and Status Buffer**

- The *transactive control and coordination system* advises a responsive resource system via this process, but it never directly controls it. A responsive resource system is not part of the *transactive control and coordination system*.

- Responsive resource systems are very diverse. Even similar systems use different approaches, practices, protocols, and standards. One might realize an opportunity for standardization in the three types of signals that will be used to advise control actions for responsive resource systems—discrete binary, discrete multilevel, and continuous.

Figure **46** is a flowchart **4600** for an exemplary “control responsive resource” process.

6.2.4 SubAppendix A: Interval Start Time Series Definition

6.2.4.1 Purpose

This section recommends a specific set of **57** Interval Start Times (IST) for use in example embodiments of the disclosed technology, including the Demonstration. The intervals range in duration from **5** minutes to **1** day. In this embodiment, the **57** ISTs define **56** intervals of varying duration, though other numbers of IST and different durations can be used.

6.2.4.2 Series of 57 Interval Start Times Defined

The first interval in a set of Interval Start Times is IST_0. While a transactive signal is being formulated, IST_0 is the next future time at which the minute hand of a clock will be at one of the **12** major divisions of an hour (e.g., on the hour, **5** minutes after the hour, **10** minutes after the hour, etc.).

The series of time intervals to be used by transactive signals during the Demonstration are as defined in Table **26**. This set of **56** intervals is easily specified, creates the same numbers of intervals, exhibits increasing coarseness into the future, and will align well with dynamic market signals that are up to **1** hour in duration. Note that a **57**th IST (e.g., IST_{56}) has been added to unambiguously define the duration of the final, **56**th interval.

One variable-length interval resides at the boundary between sets of intervals having different durations. That is, there is a variable-length interval between **5**- and **15**-minute intervals, between **15**-minute and **1**-hour intervals, between **1**- and **6**-hour intervals, and between **6**-hour and **1**-day intervals. The duration of each variable-length interval varies between the durations of the two bounding intervals, inclusive. No intervals overlap in the resulting representation of the future.

Five-minute intervals are to be used **1** hour into the future; **15**-minute intervals, **6** hours into the future; **1**-hour intervals, **1** day into the future; **6**-hour time intervals, **2** days into the future, and **1**-day intervals, **3** to **4** days into the future.

Table 26. Example Interval Time Series for use with *TIS* and *TFS*

Duration	No. Intervals	Interval Start Times
5 minutes	**12**	$IST_0, IST_0 + 0{:}05, \ldots, IST_{10} + 0{:}05$

15 minutes	20	Round(IST_{11} + 0:15)*, IST_{12} + 0:15, ..., IST_{30} + 0:15
1 hour	18	Round(IST_{31} + 1:00)*, IST_{32} + 1:00, ..., IST_{48} + 1:00
6 hours	4	Round(IST_{49} + 6:00)*, IST_{50} + 6:00, ..., IST_{52} + 6:00
1 day	2	Round(IST_{53} + 1:00:00)*, IST_{54} + 1:00:00, IST_{55} + 1:00:00
> 3 days	56 intervals	57 interval start times (IST)

* This function "Round" indicates rounding down to the next 15-minute, 1-hour, 6-hour, or 1-day interval start time. Times are indicated as dd:hh:mm, e.g., days, hours, and minutes.

The intervals of several time series that adhere to this recommendation are shown in Table 27 for several example values of IST_0.

6.2.4.3 Pseudo Code for Example IST Series

The following formula guides the calculation of the *IST* series according to the specification in Table 26. The interval start times use the notation

$$IST_n\left[dd_n, hh_n, mm_n\right], \tag{A1}$$

where "*dd*" is days, "*hh*" is hours, and "*mm*" is minutes. The value *n* refers to the sequential, ordered number of the *IST* in its series. The total number of intervals in the series is *N* = 56, where *N* is the last *n*.

$$IST \doteq \{IST_0, IST_1, IST_2, \cdots, IST_n, \cdots, IST_N\} \tag{A2}$$

The following steps and pseudo code should help standardize calculation of the members of an IST time series. The function "truncate()" indicates that the decimal parts of the result in the parentheses should be discarded.

```
(1) Calculate first element $IST_0$:
Read present time $t$
Set $IST_0 = t + 0{:}05$
Set $mm_0 = 5 * \text{truncate}(mm_0/5)$
(2) Calculate the $IST$ series for remaining 5-minute intervals:
For $n$ = 1 to 11
    Set $IST_n = IST_{n-1} + 0{:}05$
Next $n$
(3) Calculate the $IST$ series for 15-minute intervals:
Set $IST_{12} = IST_{11} + 0{:}15$
Set $mm_{12} = 15 * \text{truncate}(mm_{12}/15)$
For $n$ = 13 to 31
    Set $IST_n = IST_{n-1} + 0{:}15$
Next n
(4) Calculate the $IST$ series for 1-hour intervals:
Set $IST_{32} = IST_{31} + 1{:}00$
Set $mm_{32} = 0$
For $n$ = 33 to 49
    $IST_n = IST_{n-1} + 1{:}00$
```

```
        Next n
(5)     Calculate the $IST$ series for 6-hour intervals:
Set $IST_{50} = IST_{49} + 6:00$
Set $hh_{50} = 6 * \text{truncate}(hh_{50}/6)$
For $n = 51$ to $53$
        $IST_n = IST_{n-1} + 6:00$
  Next n
(6)     Calculate the $IST$ series for 1-day intervals:
Set $IST_{54} = IST_{53} + 1:00:00$
Set $hh_{54} = 0$
Set $IST_{55} = IST_{54} + 1:00:00$
(7)     Append the final $IST$ that indicates the end of the last 1-day interval:
Set $IST_{56} = IST_{55} + 1:00:00$
```

6.2.4.4 Example IST Series

Table 27 lists the 57 *IST* time series elements for 13 example values of IST_0. The number of intervals (56 for the Demonstration) and total described time duration, listed at the bottom of Table 27 for these examples, have been adopted as additional elements of the XML schema that has been designed for the Demonstration's transactive signals.

Table 27: Interval Start Times at Example Next Interval Start Times

Interval	#	0:00	0:05	0.10	0:15	0:30	0:45	1:00	3:00	5:00	6:00	12:00	18:00	1:00:00
5 min.	0	0:00	0:05	0:10	0:15	0:30	0:45	1:00	3:00	5:00	6:00	12:00	18:00	1:00:00
	1	0:05	0:10	0:15	0:20	0:35	0:50	1:05	3:05	5:05	6:05	12:05	18:05	1:00:05
	2	0:10	0:15	0:20	0:25	0:40	0:55	1:10	3:10	5:10	6:10	12:10	18:10	1:00:10
	3	0:15	0:20	0:25	0:30	0:45	1:00	1:15	3:15	5:15	6:15	12:15	18:15	1:00:15
	4	0:20	0:25	0:30	0:35	0:50	1:05	1:20	3:20	5:20	6:20	12:20	18:20	1:00:20
	5	0:25	0:30	0:35	0:40	0:55	1:10	1:25	3:25	5:25	6:25	12:25	18:25	1:00:25
	6	0:30	0:35	0:40	0:45	1:00	1:15	1:30	3:30	5:30	6:30	12:30	18:30	1:00:30
	7	0:35	0:40	0:45	0:50	1:05	1:20	1:35	3:35	5:35	6:35	12:35	18:35	1:00:35
	8	0:40	0:45	0:50	0:55	1:10	1:25	1:40	3:40	5:40	6:40	12:40	18:40	1:00:40
	9	0:45	0:50	0:55	1:00	1:15	1:30	1:45	3:45	5:45	6:45	12:45	18:45	1:00:45
	10	0:50	0:55	1:00	1:05	1:20	1:35	1:50	3:50	5:50	6:50	12:50	18:50	1:00:50
	11	0:55	1:00	1:05	1:10	1:25	1:40	1:55	3:55	5:55	6:55	12:55	18:55	1:00:55
15-min.	12	1:00	1:15	1:15	1:15	1:30	1:45	2:00	4:00	6:00	7:00	13:00	19:00	1:01:00
	13	1:15	1:30	1:30	1:30	1:45	2:00	2:15	4:15	6:15	7:15	13:15	19:15	1:01:15
	14	1:30	1:45	1:45	1:45	2:00	2:15	2:30	4:30	6:30	7:30	13:30	19:30	1:01:30
	15	1:45	2:00	2:00	2:00	2:15	2:30	2:45	4:45	6:45	7:45	13:45	19:45	1:01:45
	16	2:00	2:15	2:15	2:15	2:30	2:45	3:00	5:00	7:00	8:00	14:00	20:00	1:02:00
	17	2:15	2:30	2:30	2:30	2:45	3:00	3:15	5:15	7:15	8:15	14:15	20:15	1:02:15
	18	2:30	2:45	2:45	2:45	3:00	3:15	3:30	5:30	7:30	8:30	14:30	20:30	1:02:30
	19	2:45	3:00	3:00	3:00	3:15	3:30	3:45	5:45	7:45	8:45	14:45	20:45	1:02:45
	20	3:00	3:15	3:15	3:15	3:30	3:45	4:00	6:00	8:00	9:00	15:00	21:00	1:03:00
	21	3:15	3:30	3:30	3:30	3:45	4:00	4:15	6:15	8:15	9:15	15:15	21:15	1:03:15
	22	3:30	3:45	3:45	3:45	4:00	4:15	4:30	6:30	8:30	9:30	15:30	21:30	1:03:30
	23	3:45	4:00	4:00	4:00	4:15	4:30	4:45	6:45	8:45	9:45	15:45	21:45	1:03:45
	24	4:00	4:15	4:15	4:15	4:30	4:45	5:00	7:00	9:00	10:00	16:00	22:00	1:04:00
	25	4:15	4:30	4:30	4:30	4:45	5:00	5:15	7:15	9:15	10:15	16:15	22:15	1:04:15
	26	4:30	4:45	4:45	4:45	5:00	5:15	5:30	7:30	9:30	10:30	16:30	22:30	1:04:30
	27	4:45	5:00	5:00	5:00	5:15	5:30	5:45	7:45	9:45	10:45	16:45	22:45	1:04:45
	28	5:00	5:15	5:15	5:15	5:30	5:45	6:00	8:00	10:00	11:00	17:00	23:00	1:05:00
	29	5:15	5:30	5:30	5:30	5:45	6:00	6:15	8:15	10:15	11:15	17:15	23:15	1:05:15
	30	5:30	5:45	5:45	5:45	6:00	6:15	6:30	8:30	10:30	11:30	17:30	23:30	1:05:30
	31	5:45	6:00	6:00	6:00	6:15	6:30	6:45	8:45	10:45	11:45	17:45	23:45	1:05:45
1-hr.	32	6:00	7:00	7:00	7:00	7:00	7:00	7:00	9:00	11:00	12:00	18:00	1:00:00	1:06:00
	33	7:00	8:00	8:00	8:00	8:00	8:00	8:00	10:00	12:00	13:00	19:00	1:01:00	1:07:00
	34	8:00	9:00	9:00	9:00	9:00	9:00	9:00	11:00	13:00	14:00	20:00	1:02:00	1:08:00

Interval	#	0:00	0:05	0.10	0:15	0:30	0:45	1:00	3:00	5:00	6:00	12:00	18:00	1:00:00
	35	9:00	10:00	10:00	10:00	10:00	10:00	10:00	12:00	14:00	15:00	21:00	1:03:00	1:09:00
	36	10:00	11:00	11:00	11:00	11:00	11:00	11:00	13:00	15:00	16:00	22:00	1:04:00	1:10:00
	37	11:00	12:00	12:00	12:00	12:00	12:00	12:00	14:00	16:00	17:00	23:00	1:05:00	1:11:00
	38	12:00	13:00	13:00	13:00	13:00	13:00	13:00	15:00	17:00	18:00	1:00:00	1:06:00	1:12:00
	39	13:00	14:00	14:00	14:00	14:00	14:00	14:00	16:00	18:00	19:00	1:01:00	1:07:00	1:13:00
	40	14:00	15:00	15:00	15:00	15:00	15:00	15:00	17:00	19:00	20:00	1:02:00	1:08:00	1:14:00
	41	15:00	16:00	16:00	16:00	16:00	16:00	16:00	18:00	20:00	21:00	1:03:00	1:09:00	1:15:00
	42	16:00	17:00	17:00	17:00	17:00	17:00	17:00	19:00	21:00	22:00	1:04:00	1:10:00	1:16:00
	43	17:00	18:00	18:00	18:00	18:00	18:00	18:00	20:00	22:00	23:00	1:05:00	1:11:00	1:17:00
	44	18:00	19:00	19:00	19:00	19:00	19:00	19:00	21:00	23:00	1:00:00	1:06:00	1:12:00	1:18:00
	45	19:00	20:00	20:00	20:00	20:00	20:00	20:00	22:00	1:00:00	1:01:00	1:07:00	1:13:00	1:19:00
	46	20:00	21:00	21:00	21:00	21:00	21:00	21:00	23:00	1:01:00	1:02:00	1:08:00	1:14:00	1:20:00
	47	21:00	22:00	22:00	22:00	22:00	22:00	22:00	1:00:00	1:02:00	1:03:00	1:09:00	1:15:00	1:21:00
	48	22:00	23:00	23:00	23:00	23:00	23:00	23:00	1:01:00	1:03:00	1:04:00	1:10:00	1:16:00	1:22:00
	49	23:00	1:00:00	1:00:00	1:00:00	1:00:00	1:00:00	1:00:00	1:02:00	1:04:00	1:05:00	1:11:00	1:17:00	1:23:00
6-hrs.	50	1:00:00	1:06:00	1:06:00	1:06:00	1:06:00	1:06:00	1:06:00	1:06:00	1:06:00	1:06:00	1:12:00	1:18:00	2:00:00
	51	1:06:00	1:12:00	1:12:00	1:12:00	1:12:00	1:12:00	1:12:00	1:12:00	1:12:00	1:12:00	1:18:00	2:00:00	2:06:00
	52	1:12:00	1:18:00	1:18:00	1:18:00	1:18:00	1:18:00	1:18:00	1:18:00	1:18:00	1:18:00	2:00:00	2:06:00	2:12:00
	53	1:18:00	2:00:00	2:00:00	2:00:00	2:00:00	2:00:00	2:00:00	2:00:00	2:00:00	2:00:00	2:06:00	2:12:00	2:18:00
1-day	54	2:00:00	3:00:00	3:00:00	3:00:00	3:00:00	3:00:00	3:00:00	3:00:00	3:00:00	3:00:00	3:00:00	3:00:00	3:00:00
	55	3:00:00	4:00:00	4:00:00	4:00:00	4:00:00	4:00:00	4:00:00	4:00:00	4:00:00	4:00:00	4:00:00	4:00:00	4:00:00
	56	4:00:00	5:00:00	5:00:00	5:00:00	5:00:00	5:00:00	5:00:00	5:00:00	5:00:00	5:00:00	5:00:00	5:00:00	5:00:00
Totals	56	4:00:00	4:23:55	4:23:50	4:23:45	4:23:30	4:2315	4:23:00	4:21:00	4:19:00	4:18:00	4:12:00	4:06:00	4:00:00

Note 1: All times in this table are presented in the format dd:hh:mm, where "dd", "hh," and "mm" are days, hours, and minutes after time **00:00:00**.

Note 2: The row "Totals" is (**1**) the total number of intervals (not IST) being represented and (**2**) the total amount of time represented within the given time series.

6.2.5 SubAppendix C: Toolkit Function Specification Template

This example template can be completed for each toolkit function and can be posted to a common library. The following template items are used in this template:

- Function Name
- Function Version and Date
- Description—narrative description of what is to be performed or accomplished by the function
- Block Function Model—input parameters, output parameters, and actors
- Pseudo Code Implementation—parametric mathematical model or function that explains how function is implemented within the *toolkit framework.* Reference implementations that instantiate this named toolkit function should accomplish the algorithm that is laid out by this pseudo code. If that is for any reason impossible, another toolkit function should be named and described.
- Reference Implementation(s) Available—example implementation code that instantiates this function. The implementations should be referenced here in proper, complete citations.
- Future Improvements—recommend any future improvements that have been identified for this function.

6.2.6 SubAppendix C: Standard Advisory Output Control Signal

Each toolkit function that models a system of responsive assets is responsible to advise the system of assets when and to what degree it should respond. Each such toolkit function should therefore calculate a time series that states a degree of response for each current interval start time (IST). The recommendation has been summarized in Table **28**.

The following advisory signal format can be used as a standard for toolkit functions. This method accommodates advisory responses from binary (curtailed vs. normal) to several discrete levels (e.g., response level #1, response level #2, ...) to a continuum of possible responses (e.g., generate at 56% of nameplate capacity for the specified interval).

The advisory signal has been defined as a signed value to allow its application to responsive loads, responsive generation, and energy storage resources. Positive values are used when the recommended control action should increase the availability of energy by either increasing generation or by reducing load; a negative number is used when the recommended control actions should reduce generation or increase load.

The signal is quite intentionally defined in respect to a byte representation. The three most significant bits have been highlighted in Table 28 to emphasize that these bits fully represent the eight states of any asset system that has four levels of response available to it (the additional bit represents charge/discharge direction). These bits may therefore be used quite directly by simple assets or asset systems that possess limited computational capability.

1. A signed byte value is assumed (e.g., a signed 8-bit representation [-127, 127]). (For symmetry, the value -128 has not assigned. In gate logic, the use of one's complement interpretation of negative numbers accomplishes this symmetry and may be advantageous especially for controlling very simple, small assets.)

2. Positive values refer to generation [0, 127]; negative values refer to load [-0, -127].

3. The toolkit function is responsible to state a response level for each future interval, consistent with its modeled influences on transactive signals. If the asset system's number of available response levels is known with certainty at the time the toolkit function is selected, the toolkit function may prescribe a representation for each response level.

4. The asset system, or alternatively "glue" code between the toolkit function and the asset system, is responsible to interpret the advisory signal. Interpretation of the advisory signal should be made by first dividing the respective generation or load range by the number of response levels that are available from the responsive asset system. Then the asset system may determine into which of its available levels the advisory signal belongs. If a continuum of available responses exists for this asset system, the full range of the continuum should be meaningfully applied to the full nameplate rating or total population, such that the signal range is applied to the entire available resource or load range.

Example #1: Suppose toolkit load function TKLF_1.4 has been selected to model the behavior of a set of wind turbines. The behaviors of these wind turbines are not elastic and would therefore not be expected to change their operations in respect to transactive control. This toolkit function should not calculate and send any advisory control signal to the set of wind turbines. The set of wind turbines should not expect to receive any advisory control signals.

Example #2: A toolkit load function is being designed to model a system of demand responsive water heaters. The system of water heaters should be curtailed as a group. One of the outputs from the toolkit load function is designed to be a time series of advisory signals selected from the domain {0, 127}, which members represent normal and curtailed operation, respectively, for this load. (In certain implementations, and as discussed herein, a series of 56 intervals can be used, where each interval is defined by its interval start time (IST). See, e.g., Subappendix A.) The selection of the extreme advisory signals for a load having only two levels is wise because the signals will prescribe a reasonable binary response regardless of the capabilities of the asset system to which the signal is sent. The curtailable water heater system looks for signals in the ranges [0, 63] (normal operation) or [64, 127] (curtailed operation). The range [-0, -127] should be ignored (e.g., normal operation) by this responsive asset system because it can only curtail its load; it cannot increase its load in response to transactive control signals.

Example #3: A toolkit load function is created for a small residential battery storage system that has only three available response levels—fully charging, resting, and fully discharging. The function should state a time series of advisory signals to the battery system, perhaps specifying from among a set of three outputs in the set {-127, 0, 127}, which represent the three states fully charging, resting, and fully discharging, respectively. The battery system should be configured to expect one of three ranges of advisory signals [-127, -64] (charging), [-63, 63] (resting), or [64, 127] discharging.

Example #4: Another toolkit load function is created to model a battery storage storage system, but this function expects to be paired with a battery system that can operate through a continuum of responses from fully charging to fully discharging. The function creates advisory signals accordingly at any integer value in the range [-127, 127]. The battery system converts these numbers into percentages of its range of charge and discharge rates, which is done easily by dividing through by the integer 127. For example, the advisory signal value 26 is converted to 26/127, or 20.5% of its full available discharge rate.

Example #5: The small battery system of Example **#3** is paired with the toolkit load function of Example **#4**. Even though the toolkit function calculates a continuum of responses, the battery system that has only three available response levels may nonetheless respond sensibly to the advisory signal that it receives. However, because the asset's responses do not match the responses that will have been modeled by the toolkit function, the toolkit function will not correctly predict the load (and generation) that will be supplied by this battery system.

Table 28: Interpretation of Recommended Advisory Signal

	Signal	Byte	Binary Levels	Three Levels	Four Levels	...	Continuum
↑ Increase Generation / Decrease Consumption ↑	127	0111 | 1111	Level #2 of 2 (Generate / Curtail Load)	Level #3 of 3	Level #4 / 4	⋮	100.0%
	126	0111 | 1110					99.2%
	...	...					...
	96	0110 | 0000					75.6%
	95	0101 | 1111			Level #3 of 4		74.8%
	...	...					...
	85	0101 | 0101					66.9%
	84	0101 | 0100		Level #2 of 3			66.1%
	...	...					...
	64	0100 | 0000					50.4%
	63	0011 | 1111	Level #1 of 2 (Normal)		Level #2 of 4		49.6%
	...	...					...
	43	0010 | 1011					33.9%
	42	0010 | 1010		Level #1 of 3			33.1%
	...	...					...
	32	0010 | 0000					25.2%
	31	0001 | 1111			Level #1 / 4		24.4%
	...	...					...
	1	0000 0001					0.8%
	0	0000 | 0000					0.0%
↓ Decrease Generation /	-0	1111 | 1111	Level #1 of 2 (Normal)	Level #1 of 3	Level #1 / 4	⋮	-0.0%
	-1	1111 | 1110					-0.8%
	...	...					...
	-31	1110 | 0000					-24.4%

	-32	**1101 | 1111**			**Level #2 of 4**		**-25.2%**
	...	...					...
	-42	**1101 | 0101**					**-33.1%**
	-43	**1101 | 0100**		**Level #2 of 3**			**-33.9%**
	...	...					...
	-63	**1100 | 0000**					**-49.6%**
	-64	**1011 | 1111**	Level #2 of 2 (Curtail Generation / Add Load)		**Level #3 of 4**		**-50.4%**
	...	...					...
	-84	**1010 | 1011**					**-66.1%**
	-85	**1010 | 1010**		Level #3 of 3			**-66.9%**
	...	...					...
	-95	**1010 | 0000**					**-74.8%**
	-96	**1001 | 1111**			**Level #4 / 4**		**-75.6%**
	...	...					...
	-126	**1000 | 0001**					**-99.2%**
	-127	**1000 | 0000**					**-100.0%**

6.2.7 SubAppendix D: Toolkit Functions

This subappendix lists and describes example toolkit functions that can be implemented in embodiments of the disclosed technology. Two types of toolkit functions have been defined:

(1) Resource and incentive toolkit functions—used to capture the influences of energy resources and other influences upon the transactive control and coordination system's incentive signal (e.g., the TIS)

(2) Load functions—used to capture the influence of both elastic (e.g., "responsive") and inelastic loads on the transactive control and coordination system's feedback signal (e.g., the TFS).

SubAppendix B provides a template by which the toolkit functions themselves and specific reference implementations of the toolkit functions should be documented. Thereafter, these toolkit functions may be selected from a "library" of such available toolkit functions and applied at any applicable transactive nodes.

The outputs of toolkit functions constitute an interoperability boundary as the project strives to standardize the information that flows from the toolkit functions into the toolkit framework at many levels of an interoperability information stack.

6.2.8 Resource and Incentive Toolkit Functions

The example resource and incentive toolkit functions listed in Table **29** are defined and represent as instantiations of ***8****. Calculate Applicable Toolkit Resource and Incentive Functions* within the toolkit framework. Toolkit functions having the same name and number should share a common purpose and same general approach and should promise the same set of outputs into the toolkit framework. Versioning may be used for variants of these functions that differ slightly in approach, in complexity, or by the nature of expected inputs.

In Table **29**, an attempt was made to organize the functions by type and level. Following this enumeration, Function **1.1.1** would be a special implementation of Function **1.1**, which is a special implementation of Function **1.0**.

Each toolkit function should be defined by appropriate documentation following the template in SubAppendix B.

Table 29: List of Resource and Incentive Toolkit Functions

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
1.0 Imported Electrical Energy				
1.1 Non-Transactive Imported Energy	Accommodate importation of electrical energy from outside this transactive node from entities that are not themselves transactive nodes—are not participants in this transactive control and coordination	Peripheral transactive nodes that are scheduled to at times receive bulk electrical energy from outside the boundaries of this transactive control and coordination system.	Current IST time series. Historical index price or cost information about this exchange, which can inform simulation of current energy costs for this exchange of	Time series of energy exchange P_G through this corridor using the current set of IST intervals. Time series of predicted cost of energy through this corridor C_E.

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
	system.		energy. Historical energy exchanges for this corridor. Alternatively, seasonally-adjusted daily and weekly exchange schedules from which simulations may be informed and improved. Intertie exchange schedules (may be estimated from an informed simulation). Price index that represents the current delivered cost of electrical energy through this exchange corridor if such current information can be obtained. Day of week and holiday schedules.	
1.2 Transactive Imported Energy	Converts transactive signals from transactive neighbors into framework parameter outputs that are	A transactive node should restate the transactive signals that it receives in terms of toolkit framework	Current IST time series. Transactive incentive signals (TIS) from each transactive	TIS restated as energy terms C_E. TFS restated as energy terms P_G for the intervals

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
	expected by the toolkit framework.	parameters. This toolkit function is so basic that it may be treated as part of the toolkit framework.	neighbor. Transactive feedback signals (TFS) from each transactive neighbor.	during which the TFS represents imported energy.
2.0 Renewable Energy Resource				
2.1 Wind Energy	Encourage use of wind-farm-scale energy when and near where it is generated. The cost of supplying renewable energy is applied as an infrastructure cost, not as an energy cost, in order to encourage the consumption of wind energy.	Applicable to energy produced by a wind farm. May be applied to aggregated output from multiple wind farms. Use this function at transactive nodes where owners own or represent one or more wind farms. Transactive nodes that have and represent wind farm energy that is produced within their electrical boundaries.	Current IST time series. Historical wind farm power output time series, which may be used to tune and refine predictions. Actual current wind farm power output, which may be used to tune and refine predictions. Predicted wind speed and direction time series. Predicted relative humidity time series. Predicted air density time series. Predicted resource availability	Predicted average wind power P_G using intervals of the current IST time series. Infrastructure cost time series C_I using intervals of the current IST time series. (Infrastructure costs are not expected to be especially dynamic, but it is specified as a time series for consistency.)

Name, No. & Version	**Purpose**	**Where Applied**	**Inputs**	**Outputs**
			(accounts for effects of maintenance and curtailment shedding). Function that predicts wind farm power output from these conditions. Estimated amortized wind farm infrastructure expense, including operational and maintenance expenses, which estimates will be used to state the infrastructure parameter. If the costs of these specific wind farms are unavailable, secondary sources of such estimates may be used. (Infrastructure costs are probably the only costs that will be used by this function, so in some emobdiments, the infrastructure cost can be estimated from the total, long-term expense of	

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
			supplying wind energy from the resource. By doing so, the effective cost of the wind energy will be incorporated over time using a meaningful cost.)	
2.2 Solar Energy	Encourage use of solar energy when and near where it is generated. The cost of supplying renewable energy is applied as an infrastructure cost, not as an energy cost, in order to encourage the consumption of solar energy.	Applicable to medium- or large-scale solar generation. (Small solar sites may be better addressed as negative load toolkit functions, especially if such energy offsets and reduces load at this location.) Transactive nodes where owners own medium- or large-scale solar generation. Transactive nodes that have and represent the energy from solar sites within their electrical boundaries.	Current IST time series. Historical solar site power output time series, which may be used to tune and refine predictions. Actual current solar site power output, which may be used to tune and refine predictions. Predicted solar insolation time series. Predicted wind speed and direction time series. Predicted air density time series (may or may not be used). Predicted resource availability, which accounts	Predicted average solar power P_G using intervals of the current IST time series. Infrastructure cost time series C_I using intervals of the current IST time series. (Infrastructure costs are not expected to be especially dynamic, but it is specified as a time series for consistency.)

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
			for maintenance outages. Function that predicts solar power from these inputs. Estimated amortized solar site infrastructure expense, including operational and maintenance expenses, which estimates will be used to state the infrastructure parameter. If the costs of these specific solar sites are unavailable, secondary sources of such estimates may be used. Infrastructure costs are probably the only costs that will be used by this function, so in some embodiments, the infrastructure cost should be estimated from the total, long-term expense of supplying solar energy from the resource. By doing so, the	

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
			effective cost of the solar energy will be incorporated over time using a meaningful cost.	
2.3 Hydropower	TBD, based on input expected from a hydropower working group that has been asked to formulate this function. Perhaps, encourage use of hydroelectric energy when and near where it is generated. This function should at least represent federal hydropower of the region but should strive to represent all regional hydropower.	Transactive nodes that own or represent hydropower generation. Transactive nodes that have or represent hydropower generation within their electrical boundaries.	Current IST time series. Scheduled hydropower generation production targets Actual hydropower generation, if available. Day of week and holidays.	Predicted average hydropower P_G time series using the intervals of the current IST time series Predicted infrastructure cost time series C_I using intervals of the current interval start time (IST) series. (Infrastructure costs are not expected to be especially dynamic, but it is specified as a time series for consistency.)
3.0 Fossil Generation	Represent effect of fossil-fuel generation on electrical energy cost.	Transactive nodes that own or represent fossil generation. May be used for aggregated sets of fossil generation resources.	Current IST time series. Predicted cost of fuel, which may be either constant or a dynamic time series, depending on the fuel.	Predicted average generated power P_G time series using the intervals of the current IST time series. Corresponding predicted energy costs

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
		Should apply to fossil generation within the electrical boundary of a transactive node.	Generator dispatch schedule(s). Fuel heating value (probably a constant). Plant efficiency (probably a constant, but may be a function of generated power and other inputs). Outdoor temperature time series. Input feed temperature time series. Representative amortized infrastructure cost. (In some cases, the infrastructure costs will be stated as functions of many variables, including local costs of money, taxes, regulations, etc.) Function by which inputs are used to predict power output. Day of week and holidays.	of generated power C_E using the intervals of the current IST time series. Predicted infrastructure cost C_I time series using the intervals of the current IST time series. (Infrastructure cost is not expected to be especially dynamic, but it is specified as a time series for consistency.)

Name, No. & Version	**Purpose**	**Where Applied**	**Inputs**	**Outputs**
4.0 General Infrastructure Cost	Represent bulk influence of infrastructure investments on delivered cost of electrical energy where it might be impracticable to track individual infrastructure components.	Almost every transactive node could use this function.	TBD. Estimate of present infrastructure value amortized over an applicable horizon. Calculation should include effects of local influences like a utility's normal estimate of useful equipment lifetime. Estimates should be calibrated against known ways in which long-term infrastructure costs are addressed.	Infrastructure cost time series C_I.
5.0 System Constraints				
5.1 Transmission Flowgate	Discourage consumption downstream from, and encourage consumption upstream from, a flowgate transmission constraint. Costs should be grounded somehow in actual costs that would be incurred if flowgate	Transmission zone transactive nodes on either side of a flowgate.	Predicted flowgate power. Formula by which flowgate power will affect TIS each transactive node. Additional inputs may be considered for future versions, but the initial version should be kept very	Capacity cost C_C and corresponding flowgate capacity P_C.

Name, No. & Version	**Purpose**	**Where Applied**	**Inputs**	**Outputs**
	constraints were to be violated.		simple.	
5.2 Equipment and Line Constraints	Discourage consumption of energy downstream from constrained distribution equipment, including distribution lines.	Transactive nodes that are in a position to mitigate their constraints by increasing the delivered cost of energy to downstream transactive nodes. Intended to be used where constraints may be correlated to specific equipment. Does not apply to transmission flowgates.	Predicted capacity to which this function applies. Function which estimates the cost impacts of exceeding the capacity constraint.	Predicted capacity cost time series C_C and corresponding capacity time series P_C.
6.0 System Energy Losses				
6.1 Transmission Losses	Incorporate the effect of line losses on cost of delivered energy in transmission zones	Presently a low priority. Intended for application in transmission zones. May be defined and applied for major transmission across transmission zone transactive nodes.	Function by which TFS and non-transactive imported and exported power indicate long-distance transmission losses across a transmission zone. Representative fraction of transmitted power to be lost, which may be applied as a	Lost energy term of type P_G.

Name, No. & Version	**Purpose**	**Where Applied**	**Inputs**	**Outputs**
			representative resistance at a stated transmission voltage.	
6.2 Distribution Losses	Incorporate the effect of line losses on cost of delivered energy in distribution and other locations where specific lossy equipment can be identified. Reflects that the value of dissipated energy is lost.	Presently a low priority. Intended for application in the topology at locations other than transmission zones. Applied where losses may be attributed to specific equipment or systems.	Function by which TFS and non-transactive imported and exported power can be used to define energy losses in specific equipment or systems.	Lost energy term of type P_G.
7.0 Demand Charges				
7.1 BPA Demand Charges	Utility transactive node takes steps to manage peak loads that may incur demand charges. Help a utility reduce its monthly peak.	Subproject transactive nodes where owners are utilities that are subject to demand charges from BPA.	Predicted capacity to which demand charges may apply. Historical utility load during the current month, including prior peak hour. Function by which cost impact of capacity may be predicted. Day of week and holidays.	Capacity time series P_C that causes the demand charges. Capacity cost time series C_C that corresponds to the capacities. (The capacities may, or may not, also be TFS values, depending on the boundaries of a given transactive node.)

Name, No. & Version	**Purpose**	**Where Applied**	**Inputs**	**Outputs**
7.2 Seattle City Light Demand Charges	This function predicts the impact of demand charges that the Seattle City Light (SCL) will apply to the University of Washington (UW)	UW's transactive node.	SCL peak demand rate [$/kW] SCL off-peak demand rate [$/kW] Transactive Feedback Signal (TFS) [kW] Interval Start Times (ISTs) A scaling factor K by which the effect of the demand charges may be scaled.	Average power capacity P_C as defined by the Transactive Node Framework [kW]. Capacity cost C_C as defined by the Transactive Node Framework [$/kW].
8.0 Market Impacts				
8.1 Spot Market Impacts	Utility transactive node takes steps to mitigate (optimize) the predicted impacts that it will likely incur on spot markets.	Subproject transactive nodes where owners are utilities that are subject to the impacts of spot market trading.	TBD. Perhaps, predicted capacity to which spot market impacts may apply.	TBD. Perhaps, capacity time series P_C that causes the spot market impacts and capacity cost time series C_C that corresponds to the capacities. This function might use other cost time series C_O if it cannot be stated in terms of energy, capacity, or

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
				infrastructure.

6.2.9 Load Toolkit Functions

Load toolkit functions are instantiated as ***6**. Calculate Applicable Toolkit Load Functions* within the toolkit framework. The load being described by these functions may be either elastic (responsive to the *TIS*) or inelastic (not responsive to the *TIS*). These functions should not have direct influence and effect on the calculation of TIS as this transactive node; functions that will affect the formulation of TIS should be stated as resource or incentive toolkit functions.

The Demonstration attempts to define and use a minimum adequate set of load toolkit functions. Therefore, implementers should select and apply the most general function that can describe the expected behaviors. In Table **30**, an attempt was made to organize the functions by type and level. Following this enumeration, Function **1.1.1** would be a special implementation of Function **1.1**, which is a special implementation of Function **1.0**. Function **1.0** is more general that is the Function **1.1** under it.

The most general functions have been stated as

1. Bulk inelastic load—large sets of load that is not affected by the TIS

2. General event-driven demand response (DR)—sets of asset systems that are infrequently affected by the TIS. These asset systems are affected in a binary, on/off way or occasionally provide a limited number of discrete response levels. Specific examples may include distribution voltage control, water heater programs, smart appliance programs, and distributed generation.

3. General time-of-use (TOU) DR—sets of asset system that are affected by the TIS according to a daily cycle. These asset systems are affected in a binary, on/off way or occasionally provide a limited number of discrete response levels. Examples may include distribution voltage control, water heater programs, smart appliance programs, and battery storage.

General real-time (RT) DR— sets of asset systems that are affected by the TIS and employ a continuum of possible responses. Examples may include energy portals and battery storage.

Table 30: List of Load Toolkit Functions

Name, No. & Version	**Purpose**	**Where Applied**	**Inputs**	**Outputs**
1.0 Bulk Inelastic Load	Predict bulk, undifferentiated inelastic load in the most general sense.	Transactive nodes where it is preferred to predict undifferentiated bulk load. Places where specific models to predict the behaviors of differentiated load components are not possessed. Nearly every subproject could use this function.	Current IST time series. (LI_**01**) Historical load for this modeled population (LI_**02**) Present load (average power) for this population of inelastic load (LI_**03**) Predicted outdoor temperature time series (LI_**04**) Predicted insolation time series (LI_**05**) Predicted wind speed and direction time series (LI_**06**) Weekday, weekend day, and holiday indicator (LI_**08**) Typical seasonally-adjusted daily load profile	Predicted inelastic load for each current IST interval.

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
			(LI_07) Average daily load (a constant for the prediction horizon)	
1.1 Bulk Commercial Load	Predict the load of bulk inelastic commercial load. May be used to represent sets of aggregated commercial loads, even ones with diverse membership. Does not model underlying commercial buildings and processes. This model does not include elastic behaviors that would be expected to respond to a TIS.	Transactive nodes that represent inelastic electrical load from aggregated commercial loads. Most subproject transactive nodes will use this function.	Current IST time series. (LI_**01**) Historical load (LI_**02**) Actual measured load (LI_**03**) Predicted outdoor temperature time series (LI_**04**) Predicted insolation time series (LI_**05**) Predicted wind speed and direction time series (LI_**06**) Day of week and holidays (LI_**07**) Average daily load (constant during the prediction horizon) (LI_**08**) Typical daily load profile	Predicted inelastic load for each current IST interval.
1.2 Bulk Industrial Load	Predict the load of bulk industrial load types.	Transactive nodes that represent electrical load	Current IST time series. (LI_**01**) Historical	Predicted inelastic load for each current IST

Name, No. & Version	**Purpose**	**Where Applied**	**Inputs**	**Outputs**
	Does not model underlying industrial processes.	from aggregated industrial loads. This function does not require underlying industrial processes to be understood and modeled. May be applied to multiple aggregated industrial loads. Many subproject transactive nodes that include industrial loads may choose to use this function.	load (LI_**02**) Actual measured load (LI_**03**) Predicted outdoor temperature time series (LI_**04**) Predicted insolation time series (LI_**05**) Predicted wind speed and direction time series (LI_**06**) Day of week and holidays (LI_**07**) Average daily load (a constant during the prediction horizon) (LI_**08**) Typical daily load profile (LI_**09**) Fractional representation of common commercial building types	interval.
1.3 Bulk Residential Load	Predict the load of bulk residential load type. Predict load of residential feeders or	Transactive nodes that wish to represent electrical load for groups of residences	Current IST time series. (LI_**01**) Historical load (LI_**02**) Actual	Predicted inelastic load for each current IST interval.

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
	groups of residential feeders. Does not necessarily model individual residences or the underlying behaviors of homes and their occupants. Models inelastic residential load only.	like those on residential feeders. Applied to residential loads that are not responsive to the TIS (e.g., *inelastic* residential populations). Individual residences and underlying resident behaviors are not modeled. Almost every subproject transactive node is expected to use this function for its residential customers who do not respond elastically.	measured load (LI_03) Predicted outdoor temperature time series (LI_04) Predicted insolation time series (LI_05) Predicted wind speed and direction time series (LI_06) Day of week and holidays (LI_10) Number of single- and multiple- family units	
1.4 Small Wind Generator Negative Load	Predict the "negawatts" to be produced by small wind energy resources. This function is preferred where a relatively small amount of wind renewable generation	Locations that host relatively small wind generators or wind sites that primarily offset a larger electrical load.	Current IST time series. (LI_11) Historical power production time series (LI_12) Predicted wind speed and direction time series for a representative tower height	Time series output power for each IST interval. This is an *inelastic* load component because it is not a function of the TIS. No control output is sent to renewable generators. Renewable

Name, No. & Version	**Purpose**	**Where Applied**	**Inputs**	**Outputs**
	offsets load at a location. If the energy from a wind energy resource should affect TIS at this and electrically downstream locations, the energy from this resource should be incorporated with a resource and incentive toolkit function instead (See Table **29**:).		(LI_**13**) Historical wind speed and direction at a representative tower height near the wind generation (LI_**14**) Measured wind speed and direction at a representative tower height near the generation site (LI_**15**) Historical relative humidity time series (LI_**16**) Predicted relative humidity time series (LI_**17**) Historical air density time series (LI_**18**) Predicted air density time series (LI_X) Effective total cross-sectional area (LI_X) Wind conversion efficiency curve (LI_**19**) Season, or day of year (LI_**20**) Total nameplate or "typical" power	generators are not responsive to the transactive control and coordination system.

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
			capacity (LI_X) Predicted resource availability	
1.5 Small-Scale Distributed Generator Negative Load	Predict and represent "negawatts" load from one or more relatively small distributed generators that consume hydrocarbon fuels at this location. These generators are not influenced by the TIS. If the influence of a distributed generator should directly affect the TIS at a transactive node, select an appropriate source and incentive toolkit function from Table **29**:.	Locations that host relatively small fossil fuel generators that are not influenced in their operation by the TIS.	Current IST time series. (LI_**01**) Historical power production (LI_X) Resource schedule (LI_**20**) Nameplate or target power production magnitude. (LI_**06**) Day of week and holidays (LI_X) Predicted resource availability	Time series output power for each IST interval. Distributed generators of this toolkit function are not responsive to the transactive control and coordination system, but they may respond to other purposes and objectives of their owners (e.g., periodic maintenance schedules, feedstock availability). No control output is sent to these distributed generators.
1.6 Small-Scale Solar Generator Negative Load	Predict the "negawatts" to be produced by small solar energy	Locations that host relatively small solar generators that primarily offset a larger	Current IST time series. (LI_**01**) Historical power	Time series average output power for each IST interval. No control

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
	resources. This function is preferred where a relatively small amount of solar renewable generation offsets load at a location. If the energy from a solar energy resource should affect TIS at this and electrically downstream locations, the energy from this resource should be incorporated with a resource and incentive toolkit function instead (See Table **29**).	electrical load.	production (LI_??) Historical insolation time series (LI_**04**) Predicted insolation time series (LI_??) Historical wind speed and direction time series (LI_**05**) Predicted wind speed and direction time series. (LI_**15**) Historical relative humidity time series (LI_**16**) Predicted relative humidity time series. (LI_**17**) Historical air density time series (LI_**18**) Predicted air density time series (LI_**19**) Monthly typical energy (LI_**20**) Total nameplate or "typical" power capacity (LI_??) Predicted	output is sent to renewable generators. Renewable generators are not responsive to the transactive control and coordination system.

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
			resource availability (LI_??) Solar Conversion Efficiency Curve	
2.0 General Event-Driven Demand Response	Most general function for predicting the behaviors of responsive load assets that only infrequently respond. When these assets respond they change between a very limited number of available response levels. It is postulated that this function can be designed flexibly to respond to absolute or relative TIS as desired by the application.	Applicable to many responsive asset systems that conduct traditional demand response several times a month. Response may additionally define a "critical" response level for extreme conditions.	Current IST time series. Recent history (e.g., 1 day to 1 week) of TIS that may be used if relative TIS is to be tracked in a statistical sense. (LI_01) Historical load time series (LI_02) Actual measured load TIS time series. (LI_??) Device count (LI_06) Day of week and holidays (LI_08) Daily load profile (LI_28) Minimum event duration (LI_29) Promised event count or frequency that has been negotiated with customers. (LI_30) Limitations on event duration	Predicted inelastic load at for each IST interval. Predicted change in elastic load for each IST interval.

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
			that have been promised to customers.	
2.1 Commercial Event-Driven Demand Response	Represent especially the change in elastic response from commercial entities that are performing lighting, space conditioning, or other control of commercial buildings.	Asset systems such as thermostats, water heaters, and HVACs.	Current IST time series. See 1.1 Bulk Commercial Loads. The inputs that have been defined for function 1.1 Bulk Commercial Loads are again used to predict the inelastic load component of the commercial load to be modeled by this function. Additionally, the following inputs may be used to model the change in elastic load: TIS time series. Recent history (e.g., 1 day to 1 week) of TIS that may be used if relative TIS is to be tracked in a statistical sense. (LI_??) Device Count (LI_29) Promised event count or frequency that has been negotiated with	Predicted inelastic load at for each IST interval. Predicted change in elastic load for each IST interval. Predicted time series of output advisory control signals. See SubAppendix C. (Default expects two load levels specified by the domain {0, 127}). The set of output signals may be parametrically modified based on the number of available response levels, a static input.

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
			customers. (LI_**30**) Limitations on event duration that have been promised to customers. (LI_**31**) Representative unit changes in power that will occur at prescribed response levels. (LI_??) Number of response levels available from asset system.	
2.2 Event-Driven Distribution System Voltage Control	To be used where subprojects of the Demonstration have offered to modulate distribution system voltage in response to relatively extreme conditions of the TIS. This function should include the option where the degree of voltage change is affected by feedback from	Many subproject locations of the Demonstration that implement conservation voltage regulation (CVR) or voltage optimization and have offered to make system voltage responsive to the TIS.	Current IST time series. (LI_**01**) Historical load TIS time series. (LI_**32**) Present actual voltage regulation level Current IST time series (LI_**35**) Implementer's criteria concerning how often and how long voltage may be affected at each level. <u>Note</u> that this input may probably be adequately	Predicted inelastic load at for each IST interval. Predicted change in elastic load for each IST interval. Predicted time series of output advisory control signals. See SubAppendix C. (Default expects two load levels specified by the domain {**0**, **127**}). The set of output signals may be parametrically

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
	measurements of voltage at various feeder locations. Regardless, utilities should keep customer voltage within accepted ranges.		represented by input types LI_**29** and LI_**30**. (LI_**36**) Day-long hourly time series of relative fractions of load that are constant impedance, constant current, and constant power, respectively (LI_??) Number of response levels available from asset system.	modified based on the number of available response levels, a static input.
2.4 Residential Event-Driven Demand Response		Asset systems.	See **1.3** Bulk Residential Load. The inelastic residential load component may use the same inputs as were used for function **1.3** Bulk Residential Load. The following additional inputs may be used to predict changes in the elastic load component: TIS time series Current IST time series (LI_**20**) Total nameplate or "typical" power	Predicted inelastic load at for each IST interval. Predicted change in elastic load for each IST interval Predicted time series of output advisory control signals. See SubAppendix C. (Default expects two load levels specified by the domain {**0**, **127**}). The set of output signals may be parametrically modified based on the number

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
			capability (of devices to be curtailed) (LI_??) Hourly curtailable power (LI_??) Device count (LI_28) Minimum Event Duration (LI_29) Promised Event Count or Frequency (LI_30) Limitations on Curtailment Event Duration (LI_31) Representative Changes in Power at Prescribed Response Levels (LI_??) Actual Number of Times that Actuation has Already Occurred in each Relevant Time Period (LI_??) Actual duration that actuation has already occurred in each relevant time period (LI_??) Number of response	of available response levels, a static input.

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
			levels available from asset system.	
2.5 Non-Renewable Distributed Generation Event-Driven Demand Response		Asset systems.	(LI_**01**) Historical Load or Generation (LI_**02**) Actual Measured Load or Generation (LI_**06**) Day of Week and Holiday (LI_**07**) Average Daily Load or Generation (LI_**08**) Daily Load or Generation Profile (LI_**19**) Monthly Typical Energy (LI_??) Resource Schedule TIS time series (LI_??) Device Count (LI_**20**) Total nameplate or "typical" power capability (of devices to be curtailed) (LI_??) Hourly curtailable power (LI_??) Device count	Predicted inelastic load at for each IST interval. Predicted change in elastic load for each IST interval Predicted time series of output advisory control signals. See SubAppendix C. (Default expects two load levels specified by the domain {**0**, **127**}). The set of output signals may be parametrically modified based on the number of available response levels, a static input.

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
			(LI_**28**) Minimum Event Duration (LI_**29**) Promised Event Count or Frequency (LI_**30**) Limitations on Curtailment Event Duration (LI_**31**) Representative Changes in Power at Prescribed Response Levels (LI_??) Actual Number of Times that Actuation has Already Occurred in each Relevant Time Period (LI_??) Actual duration that actuation has already occurred in each relevant time period (LI_??) Number of response levels available from asset system.	
3.0 General Time-of-Use Demand Response	Most general function for predicting responsive load behaviors of	Applicable at locations that host simple DR systems that should	See function **1.0** Bulk Inelastic Load. The inputs from **1.0** Bulk Inelastic Load are also useful	Predicted inelastic load at for each IST interval. Predicted

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
	groups of devices that respond to diurnal variability in the TIS (e.g., respond to one or more daily intervals)	respond daily.	by this function for predicting the inelastic load component. Additionally, the following inputs will be useful for the prediction of changes in elastic load component: TIS time series (LI_??) Device Count (LI_**28**) Minimum Event Duration (LI_**29**) Promised Event Count or Frequency (LI_**30**) Limitations on Curtailment Event Duration (LI_**31**) Representative Changes in Power at Prescribed Response Levels (LI_??) Actual Number of Times that Actuation has Already Occurred in each Relevant Time Period (LI_??) Actual	change in elastic load for each IST interval. Predicted time series of output advisory control signals. See SubAppendix C. (Default expects two load levels specified by the domain {**0**, **127**}). The set of output signals may be parametrically modified based on the number of available response levels, a static input.

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
			duration that actuation has already occurred in each relevant time period (LI_??) Hourly Unit Expected Change in Power at Event Levels (LI_??) Number of response levels available from asset system.	
3.1 Battery Storage – Time-of-Use	Represent behaviors of battery storage systems that are engaged with a daily pattern, usually to mitigate daily peak. Battery is fully charging, fully discharging, or resting.	Locations that host usually small battery systems controlled simply on a diurnal pattern. Presently, no transactive nodes claim to be applying battery systems in this way.	(LI_**01**) Historical Load or Generation (LI_**02**) Actual Measured Load or Generation (LI_**20**) Total Nameplate or "Typical" Power Capacity (LI_??) Device Count (LI_**28**) Minimum Event Duration (LI_**29**) Promised Maximum Event Count or Frequency (LI_**30**) Limitations on Maximum Event Duration (LI_**31**) Representative	Predicted inelastic load at for each IST interval. This will normally be zero, assuming that the battery charges and discharges only for economic reasons and according to the condition of the TIS signal. Predicted change in elastic load for each IST interval. Predicted time series of output advisory control signals. See SubAppendix C. (Default expects three

Name, No. & Version	**Purpose**	**Where Applied**	**Inputs**	**Outputs**
			Changes in Power at Prescribed Response Levels (LI_??) Actual Number of Times that Actuation has Already Occurred in each Relevant Time Period (LI_??) Actual duration that actuation has already occurred in each relevant time period (LI_**41**) Predicted Resource Fractional Availability Current IST time series. TIS time series. (LI_??) Battery state of charge. (LI_??) Useful Energy Storage Capacity (LI_??) Number of response levels available from asset system.	load levels specified by the domain {-**127**, **0**, **127**}). The set of output signals may be parametrically modified based on the number of available response levels, a static input.
3.2 Commercial Time-of-Use	Represent effects of predominantl	Transactive nodes that offer	See **1.1** Bulk Commercial Loads. This	Predicted inelastic load at for each IST

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
Demand Response	y commercial lighting and space conditioning programs that respond to one or several daily peak periods.	commercial system responses for addressing daily peak.	function may use the same inputs as function 1.1. Bulk Commercial Loads as it predicts the inelastic component of its load. These additional inputs may be used to calculate the change in the elastic component of this function's load: TIS time series. (LI_??) Device Count (LI_28) Minimum Event Duration (LI_29) Promised Event Count or Frequency (LI_30) Limitations on Curtailment Event Duration (LI_31) Representative Changes in Power at Prescribed Response Levels (LI_??) Actual Number of Times that Actuation has	interval. Predicted change in elastic load for each IST interval. Predicted time series of output advisory control signals. See SubAppendix C. (Default expects two load levels specified by the domain {0, 127}). The set of output signals may be parametrically modified based on the number of available response levels, a static input.

Name, No. & Version	**Purpose**	**Where Applied**	**Inputs**	**Outputs**
			Already Occurred in each Relevant Time Period (LI_??) Actual duration that actuation has already occurred in each relevant time period (LI_??) Hourly Unit Expected Change in Power at Event Levels (LI_??) Number of response levels available from asset system.	
3.4 Residential Time-of-Use Demand Response	Predict and represent response from automated residential demand-response systems of many types that will respond approximately daily to help mitigate peak conditions. This function applied to automated responses and may accommodate customer	Applied where programmable, communicating thermostats; smart appliances, demand-response switch units, or other assets are installed in residences and where programs are designed to have these systems respond to daily peak periods. Asset systems such	See **1.3** Bulk Residential Load. This function may use the same inputs as for **1.3** Bulk Residential Load to predict the inelastic component of its load. The following additional inputs may be used to predict the change in elastic load: TIS time series. (LI_??) Device Count (LI_**28**) Minimum	Predicted inelastic load at for each IST interval. Predicted change in elastic load for each IST interval. Predicted time series of output advisory control signals. See SubAppendix C. (Default expects two load levels specified by the domain {**0**, **127**}). The set of output signals may be

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
	opt-out.	as water heater control, thermostat load control.	Event Duration (LI_29) Promised Event Count or Frequency (LI_30) Limitations on Curtailment Event Duration (LI_31) Representative Changes in Power at Prescribed Response Levels (LI_??) Actual Number of Times that Actuation has Already Occurred in each Relevant Time Period (LI_??) Actual duration that actuation has already occurred in each relevant time period (LI_??) Hourly Unit Expected Change in Power at Event Levels (LI_??) Number of response levels available from asset system.	parametrically modified based on the number of available response levels, a static input.
3.5 Time-of-	Similar to	Applicable	Current IST time	Predicted

Name, No. & Version	**Purpose**	**Where Applied**	**Inputs**	**Outputs**
Use Distribution System Voltage Control	toolkit function 2.2, except voltage may be controlled according to daily on- and off-peak periods.	where voltage is controlled at two or more levels according to the value of the TIS and other inputs and where responses of the asset have been designed to occur according to daily on-and off-peak periods.	series. Historical power consumption TIS time series. TIS threshold(s), which may further be parametrically affected. (LI_??) Number of response levels available from asset system.	inelastic load at for each IST interval. Predicted change in elastic load for each IST interval. Predicted time series of output advisory control signals. See SubAppendix C. (Default expects two load levels specified by the domain {0, 127}). The set of output signals may be parametrically modified based on the number of available response levels, a static input.
3.6 Time-of-Use Electric Vehicle Charging		Asset systems such as vehicle charging.	See 3.1 Battery Storage—Time-of-Use. This function is expected to use the same inputs as does 3.1 Battery Storage—Time-of-Use. Additionally, these inputs may be used because of the special characteristics of	Predicted inelastic load at for each IST interval. Predicted change in elastic load for each IST interval Predicted time series of output advisory control signals. See SubAppendix

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
			electric vehicles: (LI_??) Time at Which Energy Storage Should be Fully Charged (LI_??) Number of response levels available from asset system.	C. (Default expects two load levels specified by the domain {**0**, **127**}). The set of output signals may be parametrically modified based on the number of available response levels, a static input.
3.7 Non-Renewable Distributed Generation Time-of-Use Demand Response	This function predicts the response from a non-renewable distributed generator demand-response system that will respond approximately daily to help mitigate peak conditions that are evident in an incentive signal.	Asset systems.	Maximum allowed rate of change in generated power Number of response levels to be prescribed for this asset system Typical fraction of time that each response level *l* should be active during a day Minimum time duration for which an event level *l* should remain in force for this day type after it has become initiated Maximum total event duration permitted per day type and per event allowed for each event	Predicted inelastic load (generation) from this asset system Predicted average change in elastic load for each *IST* interval Predicted time series of output advisory control signals. See SubAppendix C. (Default expects two load levels specified by the domain {**0**, **127**}). The set of output signals may be parametrically modified based on the number of available response levels, a static

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
			level *l* Limitations on the minimum number of TOU events that may be called during the three major day types for each response level *l* Limitations on the maximum number of TOU events that may be called during the three major day types for each response level *l* Recent history of *TIS* Current *TIS* for future *IST* intervals Typical baseline power that is generated during UTC hour *h* of a weekday day type by this distributed generation resource Typical baseline power that is generated during hour *h* of a weekend day by this distributed generation resource Change in	input.

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
			generation that may be anticipated at each of the *L* response levels	
4.0 General Real-time Continuum Demand Response	Most general function for predicting responsive load behaviors of groups of devices that respond according to a continuum of possible responses.	Applicable at locations that host simple RT systems.	Current IST time series. Historical power consumption TIS time series. Parametric algorithm by which change in elastic load may be predicted.	Predicted inelastic load at for each IST interval. Predicted change in elastic load for each IST interval. Predicted time series of output advisory control signals. See SubAppendix C. (Default expects a continuum of advisory levels [**0**, **127**]).
4.1 Battery Storage – Real-Time	Predict and represent the response and condition of a battery system is highly responsive to the dynamic changes in the TIS and that responds using a continuum of charge and discharge	Applicable to battery storage systems that respond very dynamically to the TIS and other local conditions and provide also a continuum of charge and discharge levels. Asset systems such as Demand Shifters and	Current IST time series. Historical power consumption, generation patterns TIS time series. Parametric algorithm by which change in elastic load may be predicted. State of charge. Limitations on maximum	Predicted inelastic load at for each IST interval. Predicted change in elastic load for each IST interval. Predicted time series of output advisory control signals. See SubAppendix C. (Default expects a continuum of

Name, No. & Version	**Purpose**	**Where Applied**	**Inputs**	**Outputs**
	levels.	distribution batteries.	charge and discharge levels.	advisory levels [**-127**, **127**]).
4.2 Commercial Real-Time Demand Response	Predict and represent dynamic commercial demand-response systems that observe the full dynamics of the TIS (and other information) and dynamically respond using a continuum of possible control outcomes.	Mostly applicable to commercial space heating but may be applicable to other commercial devices that observe the full dynamics of the TIS (and other information) and respond with a continuum of possible control outcomes (e.g., temperature settings).	Current IST time series. Historical power consumption TIS time series Parametric algorithm by which change in elastic load may be predicted	Predicted inelastic load at for each IST interval. Predicted change in elastic load for each IST interval. Predicted time series of output advisory control signals. See SubAppendix C. (Default expects a continuum of advisory levels [**0**, **127**]).
4.3 Real-Time Distribution System Voltage Control		Asset systems.		Predicted inelastic load at for each IST interval. Predicted change in elastic load for each IST interval Predicted time series of output advisory control signals. See SubAppendix C. (Default expects a continuum of advisory levels

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
				[0, 127]).
4.5 Residential Real-Time Demand Response	Predict and represent responses from the most dynamic of residential demand-response system that observe the dynamics of the TIS (and other information) and automatically respond with any of a continuum of possible responses.	Applicable where residential customers possess space conditioning systems that observe the dynamics of the TIS and provide a continuum of responses. Asset systems.	Current IST time series. Historical power consumption TIS time series Parametric algorithm by which change in elastic load may be predicted Day and time of day	Predicted inelastic load at for each IST interval. Predicted change in elastic load for each IST interval. Predicted time series of output advisory control signals. See SubAppendix C. (Default expects a continuum of advisory levels [0, 127]).
5.0 General Manual or Behavioral Demand Response			(LI_??) Number of response levels available from asset system.	Predicted inelastic load at for each IST interval. Predicted change in elastic load for each IST interval. Predicted time series of output advisory control signals. See SubAppendix C. (Default expects a continuum of advisory levels [0, 127]).
5.1	Special case	Applicable	Current IST time	Predicted

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
Residential Behavioral Response to Portals or In-Home Displays	of toolkit load function 5.0 where the means of conveying demand-response information or requests to residents is either an in-home display or energy portal. An energy portal or in-home-display is a dedicated piece of equipment for the conveyance of demand-response information or advice. The actuation of energy responses is not automated by this function, but the means by which the customer is informed or advised should be automated.	where residential customers have been provided in-home displays or portals that display the TIS. Asset systems.	series. Prediction of the inelastic load output may use the same inputs as were described for function 1.0 Bulk Inelastic Load. Refer to that function. Where the load is predominantly residential, commercial, or industrial, the designer should refer to the respective functions 1.1, 1.2 or 1.3. The following additional inputs are used to predict the change in elastic load: TIS time series. (LI_??) Number of response levels available from asset system.	inelastic load at for each IST interval. Predicted change in elastic load for each IST interval. Variant #1—continuum: Current TIS signal is relayed to the portal or in-home display. Variant #2—discrete levels: Predicted time series of output advisory control signals are sent to in-home display or portal that convey discrete response levels for events or time of use periods. See SubAppendix C. (Default expects two load levels specified by the domain {0, 127}). The set of output signals may be parametrically modified based on the number of available response levels, a static

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
				input.
5.2 Residential Behavioral Response - No Portals or In-Home Displays	Predict and represent elastic response from assets that both use human decisions and action but do not use energy portals or in-home displays to convey demand-response information or requests.	Locations where humans are informed about extreme power grid events and are invited to take actions that would mitigate the events.	Current IST time series. (LI_??) Number of response levels available from asset system.	Predicted inelastic load at for each IST interval. Predicted change in elastic load for each IST interval. Variant **#1**—continuum: Current TIS signal is relayed to the portal or in-home display. Variant **#2**—discrete levels: Predicted time series of output advisory control signals are sent to in-home display or portal that convey discrete response levels for events or time of use periods. See SubAppendix C. (Default expects two load levels specified by the domain {**0**, **127**}). The set of output signals may be parametrically modified based on the number

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
				of available response levels, a static input.
5.3 Manual Commercial Demand Response		Asset Systems.	(LI_??) Number of response levels available from asset system.	Predicted inelastic load at for each IST interval. Predicted change in elastic load for each IST interval. Predicted time series of output advisory control signals. See SubAppendix C. (Default expects two load levels specified by the domain {**0**, **127**}). The set of output signals may be parametrically modified based on the number of available response levels, a static input.
5.4 Manual Non-Renewable Distributed Energy Resources Demand Response	Predictive advisory signals should be formulated and conveyed to operations personnel at Lower Valley and	Asset systems.	Current IST time series. TIS time series. (LI_**37**) Frequency or number of times that the DER may be actuated. Note:	Predicted inelastic load (generation) at for each IST interval. Predicted change in elastic load (generation) for each IST

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
	University of Washington. The operations people will then manually schedule and/or control their distributed generation resources correspondingly.		this input should be replaced by more general LI_**29**. (LI_**29**) Promised event count or frequency that have been negotiated with customer (LI_??) Number of times that actuation has already occurred in each relevant time period. (LI_??) Actual duration that actuation has already occurred in each relevant time period. Note: Should replace this input with more general LI_**30**. (LI_**30**) Limitations on curtailment event duration that have been promised to customer Note: this input should be replaced by the more general LI_**30**. Note: This should be replaced by	interval. Variant #**1**—continuum: Current TIS signal is relayed to the portal or in-home display. Variant #**2**—discrete levels: Predicted time series of output advisory control signals are sent to in-home display or portal that convey discrete response levels for events or time-of-use periods. See SubAppendix C. (Default expects two load levels specified by the domain {**0**, **127**}). The set of output signals may be parametrically modified based on the number of available response levels, a static input.

Name, No. & Version	Purpose	Where Applied	Inputs	Outputs
			LI_**20**, which shares the same meaning. (LI_**20**) Total Nameplate or "Typical" Power Capacity. (LI_**41**) Limitations on operator ability to receive and schedule responses. (LI_??) Number of response levels available from asset system.	

6.3 Appendix C – Collected Set of Example Toolkit Functions

This section introduces a variety of exemplary load and incentive functions, any one or more of which can be used in embodiments of the disclosed technology (e.g., in a toolkit library). The functions described below should not be construed as limiting in any way, and are example implementations of functions that can be used in a transactive control and coordination system. Further, the equations, tables, and subappendices in the function descriptions below will have their own independent numbering and labeling conventions. Still further, in some instances, some information may be omitted from certain functions but could be implemented by those skilled in the art.

6.3.1 Bulk Inelastic Load – N-Day Moving Window (Function 1.01)

Description:

The following is the foundation of an alternative toolkit function to **1.0** Bulk Inelastic Load. However, this functional specification can be implemented with initial measurements over only two prior days, expects less mathematical knowledge by implementers, is easily documented down to requisite steps, and, for these reasons, may be more amenable to implementation by some utility implementers.

The basic approach is as follows: For a given circuit location, pairs of electrical load and ambient temperature are measured each hour. Data from the same hour-of-day and from a comparable day type, for a window of a chosen number of days, are used to compute the coefficients of a linear model. This model is then used to predict electrical load at this location for the same future day type and hour-of-day based on the forecasted ambient temperature for the future hour.

Block Input/Output Function Model:

Inputs:

- $\{P_{d,h}, T_{d,h}\}$—[kW, °C]—paired measurements of actual electrical power (load) and ambient temperature for a given day d of a given type (weekday or weekend/holiday) and hour h of the day at a circuit location. h=**0**, **1**, ..., **23**. These measurements taken each hour allow the recursive model to become updated for the respective day type and hour-of-day.
- N—[dimensionless]—number of days in the moving window that will be used in the model formulation. Default: **10** (e.g., about two weeks of weekdays or about a month of weekend/holiday days).
- $T_{f\_d,h}$—[°C]—forecasted temperature for a given future hour-of-day h for a least the next four days (e.g., the predicted time horizon of the transactive signals). This forecasted temperature is the input to the model by which electrical power load may be predicted for a given hour-of-day and day type.

Interim Calculation Products:

- a_{0_h}, a_{1_h} —[kW, kW/°C]—a set of coefficients that model a best-fit prediction of electrical power from a forecasted ambient temperature for a given hour-of-day on a given type of day.
- A_{00_h}, A_{01_h}, A_{11_h}, b_{0_h}, b_{1_h}—set of five unique vector and matrix elements that should be stored for each hour-of-day for each day type. These elements are

updated each time a new pair of load and temperature measurements become available for the respective hour-of-day and day type.

- $\hat{P}_{d,h}$—[kW]—predicted load for each future hour for the next four days. These are the outputs from the linear model for the respective future hour-of-day and day type, given the forecasted ambient temperature for that future hour.

Outputs:

- $L_{inelastic\_n}$—[kW]—predicted load corresponding to the n^{th} interval. This is the hourly predicted load $\hat{P}_{d,h}$ allocated accordingly to each n^{th} interval.

Pseudo Code Implementation:

1. For d available measurements, calculate A_{00_h}, A_{01_h}, A_{11_h}, b_{0_h}, and b_{1_h}. At startup, two measurements (e.g., d=**2**) may be adequate. More prior measurements are preferred and may be used. It should be pointed out that singularity is unavoidable when d=**1**; the determinant of matrix **A**, as derived in Appendix A, is zero.

$$\forall h, \quad \begin{aligned} A_{00\_h} &= \min(d,N) \\ A_{01\_h} &= \sum_{i=\max(1,d-N+1)}^{d} T_{i,h} \\ A_{11\_h} &= \sum_{i=\max(1,d-N+1)}^{d} T_{i,h}^2 \\ b_{0\_h} &= \sum_{i=\max(1,d-N+1)}^{d} P_{i,h} \\ b_{1\_h} &= \sum_{i=\max(1,d-N+1)}^{d} \left(P_{i,h} \cdot T_{i,h}\right) \end{aligned} \quad , d \geq 2 \qquad (1)$$

- Note that singularity will still occur for d>**1** if $T_{i,h}$ are identical for a given h.

- This example method uses at most N x**24**x**2** data points, which are stored for each day type.

- At the implementer's discretion, equation **2** may be employed instead of equation **1**. Equation **2** modestly reduces computations. However, equation **2** uses additional data that is stored (e.g., Nx**24**x**7** compared to Nx**24**x**2**).

$$\forall h,\quad \begin{aligned}
&A_{00\_h} = \min(d, N) \\
&A_{01\_h} = \begin{cases} A^{*}_{01\_h} + T_{d,h}, & \text{if } d \le N \\ A^{*}_{01\_h} - T_{d-N,h} + T_{d,h}, & \text{otherwise} \end{cases} \\
&A_{11\_h} = \begin{cases} A^{*}_{01\_h} + T_{d,h}, & \text{if } d \le N \\ A^{*}_{11\_h} - T^{2}_{d-N,h} + T^{2}_{d,h}, & \text{otherwise} \end{cases} \\
&b_{0\_h} = \begin{cases} b^{*}_{0\_h} + P_{d,h}, & \text{if } d \le N \\ b^{*}_{0\_h} - P_{d-N,h} + P_{d,h}, & \text{otherwise} \end{cases} \\
&b_{1\_h} = \begin{cases} b^{*}_{1\_h} + P_{d,h} \cdot T_{d,h}, & \text{if } d \le N \\ b^{*}_{1\_h} - P_{d-N,h} \cdot T_{d-N,h} + P_{d,h} \cdot T_{d,h}, & \text{otherwise} \end{cases}
\end{aligned} \tag{2}$$

- A^{*}_{01}, A^{*}_{11}, b^{*}_{0}, and b^{*}_{1} are A_{01}, A_{11}, b_0, and b_1 from the preceding iteration, respectively.

2. After matrix and vector elements have been calculated by either equation **1** or equation **2**, calculate the coefficients for the linear model using equation **3**.

$$\forall h,\quad \begin{aligned}
a_{0\_h} &= \frac{A_{11\_h} b_{0\_h} - A_{01\_h} b_{1\_h}}{A_{00\_h} A_{11\_h} - A^{2}_{01\_h}} \\
a_{1\_h} &= \frac{A_{00\_h} b_{1\_h} - A_{01\_h} b_{0\_h}}{A_{00\_h} A_{11\_h} - A^{2}_{01\_h}}
\end{aligned} \tag{3}$$

3. Generate $\hat{P}$ for the upcoming four days using the linear model in equation **4**:

$$\text{for } D = \{d+1, d+2, d+3, d+4\}, \text{ and } \forall h,\quad \hat{P}_{D,h} = a_{0\_h} + a_{1\_h} \cdot T_{f\_D,h} \tag{4}$$

- The hourly standard deviation σ_h, which is potentially a useful indicator of the accuracy of and one's confidence in the hourly prediction $\hat{P}_{D,h}$, may be computed as follows:

$$\forall h,\quad \sigma_h = \sqrt{\frac{1}{\min(d,N)} \sum_{i=\max(1,d-N+1)}^{d} \left(P_{i,h} - \hat{P}_{i,h}\right)^2} = \sqrt{\frac{1}{\min(d,N)} \sum_{i=\max(1,d-N+1)}^{d} \left(P_{i,h} - \left(a_{0\_h} + a_{1\_h} \cdot T_{i,h}\right)\right)^2} \qquad (5)$$

4. Generate $L_{inelastic\_n}$ by allocating $\hat{P}_{D,h}$ to each n^{th} interval:

$$\forall n,\qquad L_{inelastic\_n} = \begin{cases} \hat{P}_{D,h}, & \text{if } n \subseteq h \\ \overline{\hat{P}_{D,h}}, & \text{if } h \subseteq n \end{cases} \qquad (6)$$

   - o $\overline{\hat{P}_{D,h}}$ is the average of all $\hat{P}_{D,h}$ corresponding to all hours h lying within n.

- Make this $L_{inelastic\_n}$ prediction available as an output of this function into the transactive node's algorithmic toolkit framework.

5. Each time a successive measurement pair becomes available, repeat starting from step **1** above.

Subappendix A: Additional Details about the Formulation

This formulation is based on a first-order polynomial (linear) model of power $\hat{P}$ as a function of temperature *T*, as shown in equation A1. This model's coefficients a_0, and a_1 are determined via a least-squares error fit to pairs of measured power and temperature. The coefficients may be used thereafter to predict power given forecasted temperatures.

$$\hat{P} = a_0 + a_1 \cdot T \qquad \text{(A1)}$$

The optimal coefficients are determined by minimization of the cost function *J* shown in equation A2. This wisely chosen cost function happens to be the statistical variance of the difference between actual measured electrical load and load that is modeled by the linear model during *N* days of a given type (weekdays, or weekends/holidays). The standard deviation is the square root of the variance. The variance and standard deviation are potentially useful indicators of the accuracy of and one's confidence in the predictions that result from this function.

$$J = \frac{1}{N}\sum_{i=1}^{N}\left(P_i - \hat{P}_i\right)^2 \qquad \text{(A2)}$$

The optimal coefficients are found by setting the partial derivatives of the cost function with respect to the two coefficients to zero, as shown in equation A3.

$$\begin{bmatrix} \dfrac{\partial J}{\partial a_0} \\ \dfrac{\partial J}{\partial a_1} \end{bmatrix} = \begin{bmatrix} -\dfrac{2}{N}\displaystyle\sum_{i=1}^{N}\left(P_i - a_0 - a_1 \cdot T_i\right) \\ -\dfrac{2}{N}\displaystyle\sum_{i=1}^{N}\left(P_i \cdot T_i - a_0 \cdot T_i - a_1 \cdot T_i^2\right) \end{bmatrix} = 0 \qquad \text{(A3)}$$

Equation A3 can be written in matrix form, as in equation A4.

$$\begin{bmatrix} N & \displaystyle\sum_{i=1}^{N} T_i \\ \displaystyle\sum_{i=1}^{N} T_i & \displaystyle\sum_{i=1}^{N} T_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \displaystyle\sum_{i=1}^{N} P_i \\ \displaystyle\sum_{i=1}^{N}\left(P_i \cdot T_i\right) \end{bmatrix} \qquad \text{(A4)}$$

The matrix is seen to be identical to its transpose. The simplified representation given in equation A5 will prove useful in referring to the various vector and matrix elements of equation A4.

$$\begin{bmatrix} A_{00} & A_{01} \\ A_{01} & A_{11} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \qquad \text{(A5)}$$

This is in the form $\mathbf{Ax} = \mathbf{b}$, the solution of which can be found by $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, as long as matrix **A** is invertible or nonsingular. Formulas exist for the inversion of a 2x2 matrix, so each coefficient may be explicitly solved for as in equation A6. This explicit representation is advantageous because it alleviates any expectation that the computational infrastructure being relied upon to conduct this function necessarily possesses any matrix solvers.

$$a_0 = \frac{A_{11}b_0 - A_{01}b_1}{A_{00}A_{11} - A_{01}^2}$$

$$a_1 = \frac{A_{00}b_1 - A_{01}b_0}{A_{00}A_{11} - A_{01}^2} \qquad \text{(A6)}$$

This method should not require a large set of training data, but some startup issues may be encountered. There is no reasonable way to predict electrical load before any comparable measurement has been made. The coefficients cannot be uniquely determined until at least two non-identical temperature measurements have been taken for a given hour of the day.

Subappendix B: Example

In this example, real power (load) P and temperature T measurements during fourteen *weekdays*—given in Table **31** and Table **32**, respectively—are used to compute $\hat{P}$, following the procedure outlined in in the Pseudo Code Implementation section. N = **10**. The resulting $\hat{P}$ is given in Table **33**, and plotted along with ±**1** standard deviation (e.g. $\pm\sqrt{J}$) and P in the set **4700** of graphs shown in Figure **47**. Notice that the "NaN" (not a number) entries on day **3** are due to the singularity of matrix **A** caused by the identical temperature points at the corresponding hours on days **1** and **2**. Figure **48** through Figure **50** comprise sets **4800**, **4900**, **5000** of graphs that show the linear least-squares error fit for each hour of the day, for days **4**, **12**, and **14**, respectively, given the measured data.

Table 31. Power *P* Measurements in kW

		d													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
h	0	126630	126380	123750	119310	108010	91850	101540	99580	110370	118090	111810	108690	94420	99760
	1	128540	127530	126080	119370	106720	90490	101250	99270	110440	115540	112920	107110	92590	99970
	2	130030	132390	128840	118230	107120	90680	102500	99460	112350	115970	114350	106530	92940	101600
	3	132300	134530	129970	119680	109430	92310	104400	100900	116400	117040	118210	108160	93430	103750
	4	136720	137780	131020	120650	110730	93910	105960	102830	120110	117440	121380	108240	95000	106250
	5	141660	143280	135970	122840	113740	96180	110190	107220	126350	120040	127690	112090	99450	111410
	6	151840	151040	144810	131840	121820	105230	119760	117030	135810	127720	135390	120230	108640	120590
	7	164120	161680	157710	142160	132860	114240	130250	129380	146520	138180	145470	129690	116230	131540
	8	166680	162390	158210	142940	134880	116610	131660	126770	151070	140760	150230	130020	115310	131170
	9	158610	156650	150760	137720	132790	116310	126940	121170	146550	138550	145140	127470	112020	121080
	10	150280	145960	144010	131050	130430	107040	119110	113870	137590	135270	135700	123850	107310	114660
	11	140770	138850	138650	120960	124670	100140	114120	107110	128370	128050	128430	120340	104290	108770
	12	132130	130430	134000	110740	120430	96160	111270	101900	120040	116560	122470	115930	103010	105390
	13	125840	125450	131130	105590	115060	96720	103900	97780	113440	109900	115470	114020	103600	103400
	14	120530	119940	130460	102400	114400	93370	102900	94950	110830	106170	114590	114710	105380	101570
	15	118960	117000	129940	102900	111120	94600	101420	92960	109080	102160	117490	116450	106720	102620
	16	116740	116360	131310	103930	111810	94570	102470	94420	109880	102600	118930	117110	110980	103650
	17	123890	121190	135200	107620	117810	102710	108120	99210	115810	108130	125210	119550	114430	111170
	18	137920	135820	141200	118540	125000	110720	119310	111320	128410	120590	131300	126230	121330	118390
	19	142510	139340	141880	122890	123340	114190	121300	118170	132660	124750	133520	126760	121940	123540
	20	142980	138900	139110	122620	119860	114130	118760	119720	134420	126250	128930	122130	116690	122810
	21	142550	137650	135470	122060	117290	111990	115170	117520	131880	124860	125790	116520	111650	120670

	22	136270	133130	130030	117390	112710	107870	109270	114110	128100	121910	120550	107950	110480	114810
	23	129740	126930	121660	112760	106290	103520	102800	111150	121650	117880	111880	99490	100620	107230

Table 32. Temperature *T* Measurements in °C

		d													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
η	0	-21.00	-22.00	-21.00	-14.00	-14.00	-4.00	-12.00	-10.00	-16.00	-17.00	-22.00	-7.00	3.00	-3.00
	1	-22.00	-22.00	-21.00	-14.00	-13.00	-4.00	-11.00	-9.00	-17.00	-14.00	-21.00	-7.00	3.00	-4.00
	2	-22.00	-23.00	-21.00	-13.00	-13.00	-4.00	-12.00	-9.00	-17.00	-14.00	-21.00	-6.00	2.00	-6.00
	3	-23.00	-23.00	-19.00	-12.00	-12.00	-4.00	-13.00	-8.00	-23.00	-12.00	-21.00	-7.00	3.00	-6.00
	4	-22.00	-22.00	-20.00	-12.00	-10.00	-5.00	-11.00	-8.00	-20.00	-11.00	-21.00	-6.00	2.00	-6.00
	5	-23.00	-24.00	-20.00	-12.00	-10.00	-5.00	-10.00	-8.00	-22.00	-11.00	-19.00	-6.00	3.00	-7.00
	6	-23.00	-24.00	-20.00	-12.00	-11.00	-8.00	-9.00	-8.00	-23.00	-11.00	-19.00	-5.00	3.00	-7.00
	7	-24.00	-23.00	-20.00	-11.00	-10.00	-7.00	-8.00	-9.00	-22.00	-11.00	-19.00	-4.00	3.00	-7.00
	8	-23.00	-22.00	-18.00	-10.00	-9.00	-9.00	-8.00	-13.00	-22.00	-11.00	-21.00	-4.00	3.00	-7.00
	9	-19.00	-19.00	-16.00	-9.00	-9.00	-8.00	-8.00	-12.00	-20.00	-10.00	-18.00	-3.00	3.00	-6.00
	10	-17.00	-16.00	-14.00	-8.00	-8.00	-7.00	-7.00	-9.00	-16.00	-8.00	-16.00	-2.00	4.00	-5.00
	11	-16.00	-14.00	-13.00	-4.00	-8.00	-5.00	-2.00	-9.00	-15.00	-8.00	-13.00	-2.00	1.00	-5.00
	12	-13.00	-13.00	-11.00	-4.00	-7.00	-3.00	-2.00	-6.00	-13.00	-8.00	-9.00	-2.00	3.00	-5.00
	13	-12.00	-11.00	-10.00	-4.00	-6.00	-2.00	-1.00	-6.00	-12.00	-5.00	-7.00	-2.00	2.00	-4.00
	14	-11.00	-8.00	-9.00	-4.00	-6.00	-1.00	-2.00	-4.00	-11.00	-4.00	-5.00	-1.00	2.00	-4.00
	15	-11.00	-7.00	-9.00	-4.00	-5.00	-1.00	-2.00	-4.00	-10.00	-4.00	-6.00	-1.00	3.00	-4.00
	16	-12.00	-7.00	-9.00	-5.00	-4.00	1.00	-3.00	-5.00	-7.00	-5.00	-6.00	-1.00	3.00	-3.00
	17	-11.00	-8.00	-9.00	-3.00	-3.00	-1.00	-3.00	-6.00	-9.00	-4.00	-6.00	0.00	3.00	-4.00
	18	-16.00	-9.00	-9.00	-8.00	-4.00	-2.00	-4.00	-7.00	-11.00	-13.00	-6.00	-1.00	3.00	-5.00
	19	-18.00	-12.00	-11.00	-8.00	-4.00	-2.00	-7.00	-13.00	-16.00	-11.00	-6.00	0.00	3.00	-6.00
	20	-19.00	-19.00	-12.00	-11.00	-5.00	-5.00	-6.00	-12.00	-16.00	-19.00	-6.00	0.00	3.00	-6.00
	21	-19.00	-19.00	-12.00	-12.00	-5.00	-5.00	-6.00	-10.00	-16.00	-20.00	-7.00	1.00	0.00	-7.00

	22	-21.00	-19.00	-15.00	-13.00	-4.00	-11.00	-10.00	-12.00	-18.00	-18.00	-7.00	2.00	-3.00	-7.00
	23	-18.00	-19.00	-14.00	-15.00	-4.00	-11.00	-11.00	-17.00	-17.00	-20.00	-6.00	2.00	-3.00	-7.00

Table 33. Predicted Load $\hat{P}$ in kW

		d												
h		1	2	3	4	5	6	7	8	9	10	11	12	13	14
	0	-	-	126630	116860	119280	97461	108369	103079	114703	116243	126716	97364	87557	98185
	1	-	-	NaN	112395	118233	95376	106179	100882	117808	110548	125738	97212	84840	98109
	2	-	-	127670	114445	118140	94987	109251	101292	119049	111531	127500	95262	86552	100584
	3	-	-	NaN	123941	120102	101061	114440	101943	134691	109605	126623	101920	90763	102800
	4	-	-	NaN	106100	116988	103511	112066	103657	132651	110065	132656	101073	90988	104233
	5	-	-	136800	121254	119314	106297	112426	107226	140265	113660	131096	104291	91067	109562
	6	-	-	154240	131262	130109	119518	115470	114200	151411	121797	139185	110797	101535	118367
	7	-	-	154360	143738	140592	130533	125652	129383	161018	133619	151356	119835	111943	128784
	8	-	-	145230	145832	141494	138080	128147	141862	163310	134415	157896	121255	114016	129418
	9	-	-	NaN	134730	137518	133002	126763	138324	159603	130209	150645	116328	112387	125733
	10	-	-	137320	131948	131114	128706	120439	126313	147527	121093	145033	109665	106991	120504
	11	-	-	137890	131747	128323	121719	105742	125792	137670	120420	131670	109383	108827	115807
	12	-	-	NaN	143520	119083	110190	100484	115130	132834	117456	119772	103365	97644	111728
	13	-	-	125060	145988	112810	102765	96465	113238	128199	107849	112711	101571	97728	107297
	14	-	-	120137	126527	111990	97488	98285	106113	128043	104100	106987	97088	95781	106263
	15	-	-	117980	119517	109447	99199	99613	106146	123478	103701	108810	95299	92736	105769
	16	-	-	116512	122828	108659	101560	105818	109718	112495	107500	109352	95750	92437	106631
	17	-	-	122090	126991	109571	109435	111212	118081	122865	110506	114634	101191	102691	111875
	18	-	-	135820	138594	125970	123013	120876	126128	131917	134944	121585	117003	117667	121233
	19	-	-	138812	139867	123367	120126	126652	137312	138238	129727	122264	118178	120110	123841
	20	-	-	NaN	138849	120765	120152	119458	129805	135371	140289	118805	114907	116592	121461
	21	-	-	NaN	135470	117430	117330	116924	123902	134394	141283	117843	110225	114585	118739

	22	-	-	126850	127799	102646	120991	116588	118679	128845	128699	109237	101337	110449	113300
	23	-	-	140980	123450	94357	115177	112747	121624	119604	124703	103275	98270	104107	106232

6.3.2 Bulk Inelastic Load – Recursive – Recursive Algorithm (Function 1.01a)

Description:

The following is the foundation of an alternative to the Bulk Inelastic Load toolkit functions **1.0** and **1.01**. However, this functional specification can be implemented with measurements over only two prior days, expects less mathematical knowledge by implementers, is easily documented down to requisite steps, and for, these reasons, may be more amenable to implementation by some utility implementers. Furthermore, unlike tookit function **1.01** that uses a moving window of a chosen number of days, this function **1.01**a is formulated as a purely recursive algorithm.

The basic approach is as follows: For a given circuit location, pairs of electrical load and ambient temperature are measured each hour. Data from the same hour-of-day and from a comparable day type are used to recursively update the coefficients of a linear model. This model is then used to predict electrical load at this location for the same future day type and hour-of-day based on the forecasted ambient temperature for the future hour.

Block Input/Output Function Model:

Inputs:

- $\{P_{d,h}, T_{d,h}\}$—[kW, °C]—paired measurements of actual electrical power (load) and ambient temperature for a given day d of a given type (weekday or weekend/holiday) and hour h of the day at a circuit location. h=**0**, **1**, …, **23**. These measurements taken each hour allow the recursive model to become updated for the respective day type and hour-of-day.
- N—[dimensionless]—number used in the recursive algorithm. The selected value of N should be greater than **2**. Default: **10** (e.g., about two weeks of weekdays or about a month of weekend/holiday days).
- $T_{f\_d,h}$—[°C]—forecasted temperature for a given future hour-of-day h for a least the next four days (e.g., the predicted time horizon of the transactive signals). This forecasted temperature is the input to the model by which electrical power load may be predicted for a given hour-of-day and day type.

Interim Calculation Products:

- a_{0_h}, a_{1_h} —[kW, kW/°C]—a set of coefficients that model a best-fit prediction of electrical power from a forecasted ambient temperature for a given hour-of-day on a given type of day.
- A_{01_h}, A_{11_h}, b_{0_h}, b_{1_h}—set of four unique vector and matrix elements that should be stored for each hour-of-day for each day type. These elements are updated each time a new pair of load and temperature measurements become available for the respective hour-of-day and day type.
- $\hat{P}_{d,h}$ —[kW]—predicted load for each future hour for the next four days. These are the outputs from the linear model for the respective future hour-of-day and day type, given the forecasted ambient temperature for that future hour.

Outputs:

- $L_{inelastic\_n}$—[kW]—predicted load corresponding to the n^{th} interval. This is the hourly predicted load $\hat{P}_{d,h}$ allocated accordingly to each n^{th} interval.

Pseudo Code Implementation:

1. For the number m of available startup measurements, calculate the initial A_{01_h}, A_{11_h}, b_{0_h}, and b_{1_h}. At startup, two unique measurements (e.g., m=2) may be adequate. More prior measurements are preferred and may be used. It should be pointed out that singularity is unavoidable when m=1; the determinant of matrix **A**, as derived in Appendix A, is zero.

$$\forall h, \quad \begin{aligned} A_{01\_h} &= \frac{1}{m}\sum_{i=1}^{m} T_{i,h} \\ A_{11\_h} &= \frac{1}{m}\sum_{i=1}^{m} T_{i,h}^{2} \\ b_{0\_h} &= \frac{1}{m}\sum_{i=1}^{m} P_{i,h} \\ b_{1\_h} &= \frac{1}{m}\sum_{i=1}^{m} \left(P_{i,h} \cdot T_{i,h}\right) \end{aligned} \quad , \ m \geq 2 \qquad \textbf{(1)}$$

2. Each time a successive measurement pair becomes available for day d, A_{01_h}, A_{11_h}, b_{0_h}, and b_{1_h} should be recursively updated as in equation **2**.

$$\forall h, \quad \begin{aligned} A_{01\_h} &= \frac{(N-1)\cdot A_{01\_h}^{*} + T_{d,h}}{N} \\ A_{11\_h} &= \frac{(N-1)\cdot A_{11\_h}^{*} + T_{d,h}^{2}}{N} \\ b_{0\_h} &= \frac{(N-1)\cdot b_{0\_h}^{*} + P_{d,h}}{N} \\ b_{1\_h} &= \frac{(N-1)\cdot b_{1\_h}^{*} + P_{d,h} \cdot T_{d.h}}{N} \end{aligned} \qquad \textbf{(2)}$$

- A_{01}^{*}, A_{11}^{*}, b_{0}^{*}, and b_{1}^{*} are A_{01}, A_{11}, b_{0}, and b_{1} from the preceding iteration, respectively.

3. Calculate the coefficients for the linear model using the equation **3**.

$$\forall h, \quad \begin{aligned} a_{0\_h} &= \frac{A_{11\_h}b_{0\_h} - A_{01\_h}b_{1\_h}}{A_{11\_h} - A_{01\_h}^{2}} \\ a_{1\_h} &= \frac{b_{1\_h} - A_{01\_h}b_{0\_h}}{A_{11\_h} - A_{01\_h}^{2}} \end{aligned} \qquad \textbf{(3)}$$

4. Generate $\hat{P}$ for the upcoming four days using the linear model in equation **4**:

$$\text{for } D = \{d{+}\mathbf{1},\ d{+}\mathbf{2},\ d{+}\mathbf{3},\ d{+}\mathbf{4}\}, \text{ and } \forall h, \qquad \hat{P}_{D,h} = a_{0\_h} + a_{1\_h} \cdot T_{f\_D,h} \qquad \textbf{(4)}$$

- If the d measurement pairs are stored and accessible, the hourly standard deviation σ_h, which is potentially a useful indicator of the accuracy of and one's confidence in the hourly prediction $\hat{P}_{D,h}$, may be computed as follows:

$$\forall h, \qquad \sigma_h = \sqrt{\frac{1}{d}\sum_{i=1}^{d}\left(P_{i,h} - \hat{P}_{i,h}\right)^2} = \sqrt{\frac{1}{d}\sum_{i=1}^{d}\left(P_{i,h} - \left(a_{0\_h} + a_{1\_h} \cdot T_{i,h}\right)\right)^2} \qquad \textbf{(5)}$$

5. Generate $L_{inelastic\_n}$ by allocating $\hat{P}_{D,h}$ to each n^{th} interval:

$$\forall n, \qquad L_{inelastic\_n} = \begin{cases} \hat{P}_{D,h}, & \text{if } n \subseteq h \\ \overline{\hat{P}_{D,h}}, & \text{if } h \subseteq n \end{cases} \qquad \textbf{(6)}$$

   - $\overline{\hat{P}_{D,h}}$ is the average of all $\hat{P}_{D,h}$ corresponding to all hours h lying within n.

- Make this $L_{inelastic\_n}$ prediction available as an output of this function into the transactive node's algorithmic toolkit framework.

6. Repeat starting from step **2** above.

Subappendix A: Additional Details about the Formulation

This formulation is based on a first-order polynomial (linear) model of power $\hat{P}$ as a function of temperature T, as shown in equation A**1**. This model's coefficients a_0, and a_1 are determined via a least-squares error fit to pairs of measured power and temperature. The coefficients may be used thereafter to predict power given forecasted temperatures.

$$\hat{P} = a_0 + a_1 \cdot T \qquad \textbf{(A1)}$$

The optimal coefficients are determined by minimization of the cost function J shown in equation A**2**. This wisely chosen cost function happens to be the statistical variance of the difference between actual measured electrical load and load that is modeled by the linear model during N days of a given type (weekdays, or weekends/holidays). The standard deviation is the square root of the variance. The variance and standard deviation are potentially useful indicators of the accuracy of and one's confidence in the predictions that result from this function.

$$J = \frac{1}{N}\sum_{i=1}^{N}\left(P_i - \hat{P}_i\right)^2 \qquad \text{(A2)}$$

The optimal coefficients are found by setting the partial derivatives of the cost function with respect to the two coefficients to zero, as shown in equation A3.

$$\begin{bmatrix} \dfrac{\partial J}{\partial a_0} \\ \dfrac{\partial J}{\partial a_1} \end{bmatrix} = \begin{bmatrix} -\dfrac{2}{N}\sum_{i=1}^{N}\left(P_i - a_0 - a_1 \cdot T_i\right) \\ -\dfrac{2}{N}\sum_{i=1}^{N}\left(P_i \cdot T_i - a_0 \cdot T_i - a_1 \cdot T_i^2\right) \end{bmatrix} = 0 \qquad \text{(A3)}$$

Equation A3 can be written in matrix form, as in equation A4.

$$\begin{bmatrix} 1 & \dfrac{1}{N}\sum_{i=1}^{N} T_i \\ \dfrac{1}{N}\sum_{i=1}^{N} T_i & \dfrac{1}{N}\sum_{i=1}^{N} T_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \dfrac{1}{N}\sum_{i=1}^{N} P_i \\ \dfrac{1}{N}\sum_{i=1}^{N}\left(P_i \cdot T_i\right) \end{bmatrix} \qquad \text{(A4)}$$

The matrix is seen to be identical to its transpose. The simplified representation given in equation A5 will prove useful in referring to the various vector and matrix elements of equation A4.

$$\begin{bmatrix} 1 & A_{01} \\ A_{01} & A_{11} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \qquad \text{(A5)}$$

This is in the form $\mathbf{Ax} = \mathbf{b}$, the solution of which can be found by $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, as long as matrix **A** is invertible or nonsingular. Formulas exist for the inversion of a 2x2 matrix, so each coefficient may be explicitly solved for as in equation A6. This explicit representation is advantageous because it alleviates any expectation that the computational infrastructure being relied upon to conduct this function necessarily possesses any matrix solvers.

$$a_0 = \frac{A_{11}b_0 - A_{01}b_1}{A_{11} - A_{01}^2}$$
$$a_1 = \frac{b_1 - A_{01}b_0}{A_{11} - A_{01}^2} \qquad \text{(A6)}$$

This method should not require a large set of training data, but some startup issues may be encountered. There is no reasonable way to predict electrical load before any comparable measurement has been made. If used non-recursively according to the formulation so far, the coefficients cannot be uniquely determined until at least two non-identical measurement pairs have been taken. Exceptions would be used to apply the method until $N > 2$.

After two non-identical measurements, the problem becomes over-determined, and the power of least-squares error fit comes into play. The question then becomes how many samples N to maintain and use. If a moving window is used, then one should store a cache of N data pairs. Furthermore, the cache should be maintained for all of the more than 24x2 sets of hours and day types that are to be modeled. The moving window approach may not be especially efficient from a computational and storage standpoint and should be avoided. A recursive approach is preferred.

In a recursive formulation, one can keep a cache of only the four most recently calculated unique vector and matrix elements (A_{01}, A_{11}, b_0, and b_1) for each day type and its hours. Each of these elements is presumed to have already been influenced by at least N prior measurements. When a new measurement pair (P_{N+1}, T_{N+1}) becomes available for this hour and hour type, one may recursively update elements as exemplified in A7 for vector element b_1. The effect of this recursive formula is that the old vector element is replaced by a new term that is a weighted sum of the old element and a new term that uses the new measurements. If N is large, the new measurements have less impact than they would if N were small.

$$b_1 = \frac{(N-1)\cdot\frac{1}{N}\sum_{i=1}^{N}(P_i \cdot T_i) + P_{N+1}\cdot T_{N+1}}{N} \qquad \text{(A7)}$$

Equation A8 more simply and generally shows how the old vector element b_1^* becomes replaced by the new one b_1. The two weighting factors are $(N\text{-}1)/N$ and $1/N$, which sums to unity.

$$b_1 = \frac{(N-1)\cdot b_1^* + P_{N+1}\cdot T_{N+1}}{N} \quad \text{(A8)}$$

Nothing prevents the application of recursive formulas of the type exemplified by A**7** and A**8** after the elements have been initialized. The first predictions may be wild and unreliable until more measurements can become incorporated into the model.

Subappendix B: Example

In this example, real power (load) *P* and temperature *T* measurements during fourteen *weekdays*—given in Table **34** and Table **35**, respectively—are used to compute $\hat{P}$, following the procedure outlined in the Pseudo Code Implementation section. The resulting $\hat{P}$ is given in Table **36**, and plotted along with ±**1** standard deviation (e.g. $\pm\sqrt{J}$) and *P* in the set **5100** of graphs in Figure **51**. Notice that the "NaN" (not a number) entries on day **3** are due to the singularity of matrix **A** caused by the identical temperature points at the corresponding hours on days **1** and **2**. Figure **52** through Figure **54** are sets **5200**, **5300**, **5400** of graphs that show the linear least-squares error fit for each hour of the day, for days **4**, **12**, and **14**, respectively, given the measured data.

Table 34. Power *P* Measurements in kW

		D													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
h	0	126630	126380	123750	119310	108010	91850	101540	99580	110370	118090	111810	108690	94420	99760
	1	128540	127530	126080	119370	106720	90490	101250	99270	110440	115540	112920	107110	92590	99970
	2	130030	132390	128840	118230	107120	90680	102500	99460	112350	115970	114350	106530	92940	101600
	3	132300	134530	129970	119680	109430	92310	104400	100900	116400	117040	118210	108160	93430	103750
	4	136720	137780	131020	120650	110730	93910	105960	102830	120110	117440	121380	108240	95000	106250
	5	141660	143280	135970	122840	113740	96180	110190	107220	126350	120040	127690	112090	99450	111410
	6	151840	151040	144810	131840	121820	105230	119760	117030	135810	127720	135390	120230	108640	120590
	7	164120	161680	157710	142160	132860	114240	130250	129380	146520	138180	145470	129690	116230	131540
	8	166680	162390	158210	142940	134880	116610	131660	126770	151070	140760	150230	130020	115310	131170
	9	158610	156650	150760	137720	132790	116310	126940	121170	146550	138550	145140	127470	112020	121080
	10	150280	145960	144010	131050	130430	107040	119110	113870	137590	135270	135700	123850	107310	114660
	11	140770	138850	138650	120960	124670	100140	114120	107110	128370	128050	128430	120340	104290	108770
	12	132130	130430	134000	110740	120430	96160	111270	101900	120040	116560	122470	115930	103010	105390
	13	125840	125450	131130	105590	115060	96720	103900	97780	113440	109900	115470	114020	103600	103400
	14	120530	119940	130460	102400	114400	93370	102900	94950	110830	106170	114590	114710	105380	101570
	15	118960	117000	129940	102900	111120	94600	101420	92960	109080	102160	117490	116450	106720	102620
	16	116740	116360	131310	103930	111810	94570	102470	94420	109880	102600	118930	117110	110980	103650
	17	123890	121190	135200	107620	117810	102710	108120	99210	115810	108130	125210	119550	114430	111170
	18	137920	135820	141200	118540	125000	110720	119310	111320	128410	120590	131300	126230	121330	118390
	19	142510	139340	141880	122890	123340	114190	121300	118170	132660	124750	133520	126760	121940	123540
	20	142980	138900	139110	122620	119860	114130	118760	119720	134420	126250	128930	122130	116690	122810
	21	142550	137650	135470	122060	117290	111990	115170	117520	131880	124860	125790	116520	111650	120670
	22	136270	133130	130030	117390	112710	107870	109270	114110	128100	121910	120550	107950	110480	114810

	23	129740	126930	121660	112760	106290	103520	102800	111150	121650	117880	111880	99490	100620	107230

Table 35. Temperature *T* Measurements in °C

		d													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
h	0	-21.00	-22.00	-21.00	-14.00	-14.00	-4.00	-12.00	-10.00	-16.00	-17.00	-22.00	-7.00	3.00	-3.00
	1	-22.00	-22.00	-21.00	-14.00	-13.00	-4.00	-11.00	-9.00	-17.00	-14.00	-21.00	-7.00	3.00	-4.00
	2	-22.00	-23.00	-21.00	-13.00	-13.00	-4.00	-12.00	-9.00	-17.00	-14.00	-21.00	-6.00	2.00	-6.00
	3	-23.00	-23.00	-19.00	-12.00	-12.00	-4.00	-13.00	-8.00	-23.00	-12.00	-21.00	-7.00	3.00	-6.00
	4	-22.00	-22.00	-20.00	-12.00	-10.00	-5.00	-11.00	-8.00	-20.00	-11.00	-21.00	-6.00	2.00	-6.00
	5	-23.00	-24.00	-20.00	-12.00	-10.00	-5.00	-10.00	-8.00	-22.00	-11.00	-19.00	-6.00	3.00	-7.00
	6	-23.00	-24.00	-20.00	-12.00	-11.00	-8.00	-9.00	-8.00	-23.00	-11.00	-19.00	-5.00	3.00	-7.00
	7	-24.00	-23.00	-20.00	-11.00	-10.00	-7.00	-8.00	-9.00	-22.00	-11.00	-19.00	-4.00	3.00	-7.00
	8	-23.00	-22.00	-18.00	-10.00	-9.00	-9.00	-8.00	-13.00	-22.00	-11.00	-21.00	-4.00	3.00	-7.00
	9	-19.00	-19.00	-16.00	-9.00	-9.00	-8.00	-8.00	-12.00	-20.00	-10.00	-18.00	-3.00	3.00	-6.00
	10	-17.00	-16.00	-14.00	-8.00	-8.00	-7.00	-7.00	-9.00	-16.00	-8.00	-16.00	-2.00	4.00	-5.00
	11	-16.00	-14.00	-13.00	-4.00	-8.00	-5.00	-2.00	-9.00	-15.00	-8.00	-13.00	-2.00	1.00	-5.00
	12	-13.00	-13.00	-11.00	-4.00	-7.00	-3.00	-2.00	-6.00	-13.00	-8.00	-9.00	-2.00	3.00	-5.00
	13	-12.00	-11.00	-10.00	-4.00	-6.00	-2.00	-1.00	-6.00	-12.00	-5.00	-7.00	-2.00	2.00	-4.00
	14	-11.00	-8.00	-9.00	-4.00	-6.00	-1.00	-2.00	-4.00	-11.00	-4.00	-5.00	-1.00	2.00	-4.00
	15	-11.00	-7.00	-9.00	-4.00	-5.00	-1.00	-2.00	-4.00	-10.00	-4.00	-6.00	-1.00	3.00	-4.00
	16	-12.00	-7.00	-9.00	-5.00	-4.00	1.00	-3.00	-5.00	-7.00	-5.00	-6.00	-1.00	3.00	-3.00
	17	-11.00	-8.00	-9.00	-3.00	-3.00	-1.00	-3.00	-6.00	-9.00	-4.00	-6.00	0.00	3.00	-4.00
	18	-16.00	-9.00	-9.00	-8.00	-4.00	-2.00	-4.00	-7.00	-11.00	-13.00	-6.00	-1.00	3.00	-5.00
	19	-18.00	-12.00	-11.00	-8.00	-4.00	-2.00	-7.00	-13.00	-16.00	-11.00	-6.00	0.00	3.00	-6.00
	20	-19.00	-19.00	-12.00	-11.00	-5.00	-5.00	-6.00	-12.00	-16.00	-19.00	-6.00	0.00	3.00	-6.00
	21	-19.00	-19.00	-12.00	-12.00	-5.00	-5.00	-6.00	-10.00	-16.00	-20.00	-7.00	1.00	0.00	-7.00
	22	-21.00	-19.00	-15.00	-13.00	-4.00	-11.00	-10.00	-12.00	-18.00	-18.00	-7.00	2.00	-3.00	-7.00

	23	-18.00	-19.00	-14.00	-15.00	-4.00	-11.00	-11.00	-17.00	-17.00	-20.00	-6.00	2.00	-3.00	-7.00

Table 36. Predicted Load $\hat{P}$ in kW

		d													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
h	0	-	-	126630	124191	119498	96626	108269	102805	114650	116329	127101	96474	85579	98352
	1	-	-	NaN	112395	118196	94929	105910	100573	117443	110287	125689	96505	83045	98352
	2	-	-	127670	112362	117984	94487	108956	100904	118733	111250	127527	94263	84303	101254
	3	-	-	NaN	123941	120116	101062	113747	101384	133700	109346	127434	101173	86802	103607
	4	-	-	NaN	106100	116809	103093	111749	103192	132447	109778	133525	100222	87433	104971
	5	-	-	136800	121561	119302	106235	112052	106958	139419	113423	131534	103789	88863	110968
	6	-	-	154240	134747	130419	119543	115100	114163	150572	121766	139918	109873	98412	119912
	7	-	-	154360	142393	140515	130391	125117	129120	159899	133373	151689	118670	109277	130146
	8	-	-	145230	143146	141225	137822	127300	141211	163014	133587	158918	118595	109151	130867
	9	-	-	NaN	134730	137507	132860	126121	137835	160160	129541	152236	113491	107649	127317
	10	-	-	137320	127838	130750	128511	119661	125622	146717	120059	145536	107769	103493	122319
	11	-	-	137890	130499	128391	121910	105218	125576	138497	120412	132764	107940	107169	117351
	12	-	-	NaN	143520	118649	110789	100537	114747	131530	117284	119690	102612	97249	114317
	13	-	-	125060	137416	112662	104182	97282	112642	126957	107679	112801	101343	97428	110096
	14	-	-	120137	121422	113458	103761	100347	106438	124786	104472	107208	98929	98885	109909
	15	-	-	117980	116726	111867	105021	102011	106541	120376	104146	108635	97775	97505	109736
	16	-	-	116512	118213	112656	108185	106796	109286	111196	107311	108670	100318	100467	109828
	17	-	-	122090	119859	111253	111212	111197	116548	121170	110107	114007	102980	105753	115742
	18	-	-	135820	136638	129992	126212	123068	126902	131885	134883	122509	117508	116577	125388
	19	-	-	138812	138298	127833	122911	127317	136713	139585	130864	122535	117216	118237	127802
	20	-	-	NaN	138849	120367	120023	119184	129288	135266	140450	118000	113418	114014	124063
	21	-	-	NaN	135470	116724	117127	116668	123607	134221	141567	117392	107961	113130	121303

	22	-	-	126850	126981	103428	121141	116431	118619	129812	129610	107833	98446	109999	115050
	23	-	-	140980	124582	95629	115900	113221	123598	122204	128027	101939	95493	103983	108230

6.3.3 Transactive Imported Energy (Function 1.2)

Description:

Converts transactive signals from transactive neighbors into framework parameter outputs that are expected by the toolkit framework.

Application: A transactive node typically should restate the transactive signals that it receives in terms of toolkit framework parameters.

This toolkit function is so basic that it may be treated as part of the toolkit framework.

Block Input/Output Function Model:

Inputs: Current IST time series.

Transactive incentive signals (TIS) from each transactive neighbor.

Transactive feedback signals (TFS) from each transactive neighbor.

Outputs: TIS restated as energy terms C_E.

TFS restated as energy terms P_G for the intervals during which the TFS represents imported energy.

6.3.4 Small Wind Generator Negative Load (Function 1.4)

Description:

This function is to predict the power to be produced by small wind energy resources. This function is preferred where a relatively small amount of wind renewable generation offsets load at a location.

If the energy from a wind energy resource should directly affect the transactive incentive signal (*TIS*) at this location and electrically downstream locations, the energy from this resource should be incorporated with the Wind Energy resource and incentive toolkit function instead.

This function applies to locations that host relatively small wind generators or wind sites that primarily offset a larger electrical load.

Block Input/Output Function Model:

Inputs:

- $\{u_k\}$—[m/s]—Time series of predicted wind speed for a future interval k, for the upcoming four days (time horizon of transactive signals), based on wind speed data recorded at a height h, at or close to the location under consideration. Although granular data is desired, this function is formulated to work with any available data interval.
- h—[m]—Height at which wind speed is predicted.
- Ψ—[unitless]—Wind turbine manufacturer and model information to be chosen from this preliminary text enumeration:
    - Honeywell WT6500
    - Windspire 1.2
    - Home Energy Americas V200
    - Skystream
    - Bergey Excel 10
    - Urban Green Energy UGE-4K
    - Tangarie Gale 10
    - WePower Falcon 5.5
    - Wing-Power Prototype

This enumeration should be augmented whenever a new wind turbine is to be considered.

- m—[count]—Number of wind turbines.
- h_{hub}—[m]—wind turbines' representative hub height.
- K_n—[unitless]—Time series availability fraction, e.g. fraction of turbines or wind site that is predicted to be online during each n^{th} IST interval, where $n = 0, 1, \ldots, 55$. Wind generation may be limited or entirely unavailable due to maintenance schedules and other reasons.

Interim Calculation Products:

- $\{u_{hub,k}\}$—[m/s]—Time series of predicted wind speed for a future interval k at the wind turbines' representative hub height h_{hub}.

Output:

- $\{L_n\}$—[kW]—Time series of average power to be produced by wind turbine(s) for each future n^{th} IST interval.

Pseudo Code Implementation:

1. Restate inputs in the units specified in previous section, if necessary.
2. Compute $\{u_{hub,k}\}$:

- Based on the wind profile power law relationship (Elliot 1986):

$$\forall k, u_{hub,k} = \left(\frac{h_{hub}}{h}\right)^{\alpha} \cdot u_k \quad (1)$$

  - α—[unitless]—An empirically derived constant for the location of the wind turbine(s). If empirical derivation is not possible, 1/7 may be used as an approximation.

The implementer may choose to use a different approach/relationship, if deemed more appropriate/accurate.

3. Generate $\{L_n\}$:

- For the given Ψ input, generate $\{L_k\}$ by looking up, from Table **37**, an L_k corresponding to each $u_{hub,k}$:

Table 37. Lookup table for wind turbine power output at a given wind speed

	L [kW]								
u [m/s]	**Honeywell WT6500**	**Windspire 1.2**	**ome Energy Americas V200**	**Skystream 3.7**	**Bergey Excel 10**	**Urban Green Energy UGE-4K**	**Tangarie Gale 10**	**WePower Falcon 5.5**	**Wing-Power *Prototype***
1.0	0.009	0	0	0	0.020	0	0	0	*0*
1.5	0.015	0	0	0	0.030	0	0	0	*0*
2.0	0.025	0	0	0	0.080	0	0.333	0	*0*
2.5	0.038	0	0	0	0.105	0.041	0.617	0	*0*
3.0	0.048	0	0.005	0	0.159	0.082	0.833	0.066	*0.029*
3.5	0.074	0	0.014	0.024	0.254	0.123	1.167	0.166	*0.072*
4.0	0.103	0.030	0.025	0.072	0.382	0.185	1.417	0.298	*0.130*
4.5	0.128	0.065	0.040	0.144	0.636	0.247	1.667	0.464	*0.202*
5.0	0.171	0.115	0.059	0.220	0.891	0.309	2.167	0.633	*0.276*
5.5	0.209	0.160	0.082	0.336	1.209	0.391	2.667	0.895	*0.391*
6.0	0.251	0.220	0.100	0.456	1.527	0.514	3.083	1.127	*0.492*
6.5	0.285	0.283	0.122	0.600	2.036	0.658	3.583	1.358	*0.593*
7.0	0.333	0.350	0.145	0.744	2.482	0.823	4.167	1.590	*0.694*
7.5	0.392	0.425	0.178	0.936	2.991	0.988	4.833	1.855	*0.809*
8.0	0.457	0.525	0.225	1.104	3.627	1.193	5.500	2.087	*0.911*
8.5	0.500	0.610	0.285	1.320	4.391	1.440	6.167	2.319	*1.012*
9.0	0.583	0.750	0.372	1.542	5.218	1.708	6.833	2.584	*1.128*

9.5	0.651	0.880	0.460	1.780	6.109	2.058	7.667	2.916	1.272
10.0	0.714	1.025	0.552	2.000	6.936	2.366	8.417	3.346	1.460
10.5	0.793	1.138	0.642	2.136	7.891	2.675	9.250	3.877	1.692
11.0	0.888	1.188	0.733	2.254	8.909	3.086	10.000	4.340	1.894
11.5	0.981	1.200	0.822	2.325	10.055	3.601	11.000	4.771	2.082
12.0	1.069	1.200	0.900	2.372	10.945	4.012	12.000	5.102	2.226
12.5	1.172	1.175	1.005	2.396	11.709	4.074	13.083	5.300	2.313
13.0	1.250	1.138	1.100	2.410	12.091	4.000	14.167	5.400	2.356
13.5	1.357	1.000	1.214	2.410	12.345	4.000	15.167	5.500	2.400
14.0	1.466	0.300	1.325	2.396	12.473	4.000	16.417	5.500	2.400

The information in Table **37** is plotted in graph **5500** of Figure **55**. Table **37** is based on information available in the datasheets or brochures of these wind turbines. The powers given for **32** m/s are to be used for speeds beyond **32** m/s. The datasheet of this wind turbine claims that it does not have a cut-out wind speed. Therefore, the power output has been extrapolated beyond **20** m/s. However, the extrapolated data should be replaced if more accurate data is available. This wind turbine has a cut-out speed of **30** m/s, but power output data between **20** and **30** m/s is missing in its datasheet. This data has been extrapolated here, but should be replaced if more accurate data is available. No cut-out speed information is given in the datasheet of this wind turbine. It is assumed that there is no cut-out speed and the power output has been extrapolated beyond **14** m/s. This data has been extrapolated here, but should be replaced if more accurate data is available. There is no datasheet for this prototype wind turbine. Given its similarities with the WePower Falcon **5.5**, its power versus wind speed data is assumed to be a scaled version of the Falcon **5.5**. However, this data should be replaced by either empirical data or such data from a different source.

- Allocate $\{L_k\}$ to each n^{th} interval, scale by m, and multiply by K to generate $\{L_n\}$:

$$\forall n, L_n = \begin{cases} m \cdot K_n \cdot L_k, & \text{if } n \subseteq k \\ m \cdot K_n \cdot \overline{L_k}, & \text{if } k \subseteq n \end{cases} \qquad \textbf{(2)}$$

  - $\overline{L_k}$ —[kW]—weighted-average of all L_k within n.

- Make this $\{L_n\}$ prediction available as an output of this function into the transactive node's algorithmic toolkit framework.

6.3.5 Small-Scale Solar Generator Negative Load (Function 1.6)

Description:

This function is to predict the power to be produced by small solar energy resources. This function is preferred where a relatively small amount of solar renewable generation offsets load at a location.

If the energy from a solar energy resource should directly affect the transactive incentive signal (*TIS*) at this location and electrically downstream locations, the energy from this resource should be incorporated with the Solar Energy resource and incentive toolkit function instead.

This function applies to locations that host relatively small solar generators or solar sites that primarily offset a larger electrical load.

Block Input/Output Function Model:

Inputs:

- $\{GTI_k\}$—[kW/m^2]—Time series of predicted Global Tilted Irradiance (*GTI*) for a future interval *k*, for the upcoming four days (time horizon of transactive signals), based on solar irradiance data recorded at or close to the location under consideration. ($GTI = DNI \cdot \cos(\theta_i) + DIF \cdot (1 - \beta/180°)$, where *DNI* is the Direct Normal Irradiance, *DIF* the Diffuse Horizontal Irradiance, β the inclination angle of the tilted plate, and θ_i the angle between *DNI* and the normal of the tilted plate. *DNI* and *DIF* are the actual data measured at the location under consideration.
Furthermore, $\cos(\theta_i) = \cos\beta \cdot \cos Z + \sin\beta \cdot \sin Z \cdot \cos(\theta - \psi)$, where θ and Z are the sun's azimuth and zenith, respectively. Note that there are known equations to compute θ and Z throughout the day, every day, at a given latitude.) Although

granular data is desired, this function is formulated to work with any available data interval. The *GTI* represents the effective irradiance normal to a tilted surface. For a fixed flat-plate photovoltaic (PV) collector, the computation of *GTI* is, therefore, dependent on its inclination angle β and azimuth ψ, as defined below. Note also that *GTI* may not be shared amongst solar generators unless they have the same inclination and azimuth. For a solar-tracking collector or concentrating collector, the computation of *GTI* should assume that the normal of the solar collector is in line with the Direct Normal Irradiance (*DNI*).

- β—[°]—Inclination angle of the fixed flat-plate PV collector. This is **0**° for systems laying horizontal to the ground. This is not required for solar-tracking collectors, including concentrating collectors.

- ψ—[°]—Azimuth of the fixed flat-plate PV collector. This is **180**° for systems facing due south. This is not required for solar-tracking collectors, including concentrating collectors.

- A—[m^2]—Effective surface area of the solar collector.

- η—[%]—Overall conversion efficiency of the solar energy resource, e.g. from the incident solar power (e.g., $GTI \cdot A$) to the usable alternating current (AC) power. This should be the product of the efficiencies of the solar collector and its power converter, and, if possible, should include conduction losses. The implementer may choose to model this overall efficiency as a function of power. While the efficiency of the solar collector may be constant at different power levels, the efficiency of the power converter varies. The efficiency versus power curve of the converter is sometimes published in its datasheet. Conduction losses also vary with power, but may be harder to quantify and model.

- m—[count]—Number of such solar energy resources that is being modeled by this function.

- K_n—[unitless]—Time series availability fraction, e.g. fraction of solar energy resources or solar site that is predicted to be online during each n^{th} IST interval, where n = **0**, **1**,..., **55**. Solar generation may be limited or entirely unavailable due to maintenance schedules and other reasons.

Interim Calculation Product:

- $\{L_k\}$—[kW]—Time series of average power to be produced by one solar energy resource for each future interval k.

Output:

- $\{L_n\}$—[kW]—Time series of average power to be produced by the solar energy resource(s) for each future n^{th} interval.

Pseudo Code Implementation:

1. Restate inputs in the units specified in previous section, if necessary.
2. Generate $\{L_k\}$:

- For each future interval k, compute the average power $\{L_k\}$ to be produced by one solar energy resource:

$$\forall k, L_k = GTI_k \cdot A \cdot \eta \qquad \textbf{(2)}$$

3. Generate $\{L_n\}$:

- Allocate $\{L_k\}$ to each n^{th} interval, scale by m, and multiply by K_n to generate $\{L_n\}$:

$$\forall n, L_n = \begin{cases} m \cdot K_n \cdot L_k, & \text{if } n \subseteq k \\ m \cdot K_n \cdot \overline{L_k}, & \text{if } k \subseteq n \end{cases} \qquad \textbf{(3)}$$

  - $\overline{L_k}$ —[kW]—weighted-average of all L_k within n.

- Make this $\{L_n\}$ prediction available as an output of this function into the transactive node's algorithmic toolkit framework.

6.3.6 General Event-Driven Demand (Function 2.0)

Description:

This is a very general function for predicting the behaviors of responsive load assets that only infrequently respond to events that may be identified from an incentive signal. When these assets respond, they transition to a limited number of available response levels. This general function may serve as a template for functions that are more narrowly targeted to specific responsive asset systems. This function has been written at such a high level that it will not likely be referenced and used for any asset system. But this function description will be valuable guidance to those who design more specific functions for more specific asset systems.

This function can respond to absolute or relative TIS as desired by an application.

This function applies to many responsive asset systems that conduct traditional demand response several times a month. Response may additionally define a "critical" response level for extreme conditions.

Block Input/Output Function Model:

Inputs:

Current IST time series.

TIS time series. Recent history (e.g., 1 day to 1 week) of TIS that may be used if relative TIS is to be tracked in a statistical sense.

Numbers of assets in this asset system population that may be used to scale this function.

Typical daily or weekly inelastic load profile for the asset systems that are being predicted by this function. This profile is a starting point for predicting the inelastic load component.

Outputs:

Predicted inelastic load at for each IST interval.

Predicted change in elastic load for each IST interval.

Predicted advisory control signal for this asset system.

Pseudo Code Implementation:

Inelastic load component. This algorithm will not predict an inelastic load component. Inelastic load components are better addressed by inelastic load functions that have been defined.

Elastic load component. This algorithm will calculate (**1**) predicted change in electrical load in response to the incentive signal (e.g., the asset's elasticity), (**2**) "events" during which an asset is predicted to respond, and (**3**) the predicted advisory control signal that will be sent to this elastic asset system.

Predicted Change in Electrical Load in Response to the Incentive Signal. To predict a change in energy that can result from this asset system during events, this function should model the consumption (or generation) of energy by this asset system. At least two approaches can be accommodated: (**1**) An explicit time-series load shape may be used to represent the responsive load (or generation) from this asset system. Alternatively, (**2**) A dynamic model of this asset system may be simulated to predict the effect that an event will have on the asset system. These approaches will be compared by discussing how each one could be used to predict the change in electrical load that could be had from a set of residential tank water heaters.

Explicit Time-series Load Shape. The average electrical load consumed during each hour of a day by a residential **40**-gallon tank electric water heater may be obtained. In some cases, regional and seasonal variations may be found. See (Hammerstrom **2007**, Figure **4.18**) for example. The load curves represent the average power that is expected to be consumed by an electric water heater at any time of the day. In many cases, splines will allow such load curves to be very efficiently stored and reproduced. The number of water heaters in the asset system population is a scaling factor that may be used to predict the entire consumption by this population of water heaters. If an event were to occur and cause this population of water heaters to become curtailed, the change in energy consumption by these water heaters would be predicted well by knowing the number of water heaters, the representative load curve for a single water heater, and the time and duration of the event.

Dynamic Asset System Model. The same population of electric water heaters may be more rigorously modeled using a physics-based model of a water heater. In this case, one could input typical residential hot water consumption instead of an electrical load curve. As water is consumed, hot water leaves the water tank, cold water enters the water tank, and the temperature of the water in the tank decreases. The modeled thermostat turns on the electrical heating element and heats the water at a rate that is determined by the power rating of a heating element. If the model being used is accurate, the resulting electrical load curve would also be accurate on a "typical" day.

However, if a curtailment is predicted, the response of the dynamic water heater model can predict secondary effects that could not have been modeled otherwise. After a period of electrical curtailment, the water in the tank will have become relatively cold. When the curtailment period ends, additional energy is then used to reheat the cool, stored water to the desired temperature. A rebound effect is thereby predicted at the conclusion of the curtailment event.

Events during which this Asset is Predicted to Respond. The capabilities and availability of the modeled asset system determine a set of incentive thresholds that should be managed by this function. A threshold may be a function of time. An asset system that has only two modes of operation (e.g., normal and curtailed) will define only one threshold. Generally, an asset system that has m modes of operation should define m-1 thresholds. The resulting thresholds, in turn, define m-1 levels of response for an asset system. (The "Normal" mode of operation is indeed a mode of operation, but it is usually not considered a response level.) "Events" occur any time that the predicted incentive signal exceeds a defined threshold to invoke one of the levels of response that is a feature of this asset system.

The availability of asset systems that are responsive either on an event-based or time-of-use basis may be predicted if limitations on the numbers and durations of events are stated. For example, a utility might have contracted with its customers that a responsive asset will not become curtailed by the utility more often than four times per calendar month and that none of these curtailments will not endure for more than 2 hours.

Over time, statistical distributions and correlations emerge from the dynamic behaviors of the incentive signal. This function may incorporate the behaviors of past historical incentive signals and the predicted incentive signals as these statistics are being compiled. This function may thereafter refer to such statistics to evaluate and predict where a threshold should be placed to initiate just fewer than the allowed number of events and just less than the allowed duration of events. Automated event-driven demand response will be attempting to identify events within monthlong durations, so these functions should use the actual incentive signal (not its statistical average), or it should track the statistical average of the incentive signal quite slowly in comparison with that duration.

Predicted Advisory Control Signal. Once events have been predicted, the predicted advisory control signal may be stated, aligned in time with the predicted events, according to the standardized method described in the appendix entitled "Standard Advisory Output Control Signal". In the referenced method, the capabilities of this asset system and, in some cases, the severity of an event determine which integer member of a signed byte signal will be sent to the asset system. (The domain of relevant advisory control signals will be relatively small for functions that are formulated for specific asset systems.)

6.3.7 Incentive function – Wind energy (Function 2.1)

Description:

This function addresses wind power generation and is to be applied at transactive nodes which have and represent wind farm energy that is produced within or near their electrical boundaries to encourage the use of wind energy when and near where it is generated. This function is applicable to energy produced by a wind farm or may be applied to aggregated output from multiple wind farms.

The cost of supplying the wind energy generated is applied as an infrastructure cost, in units of cost per time, consistent with the Transactive Node Framework. For simplicity, the infrastructure cost will use the $2155/kW capacity-weighted average installed cost for a wind farm. The infrastructure cost of a wind farm can thus be estimated if its capacity is known. This cost shall then be spread over the lifetime *T* of the wind farm.

Note that this calculation typically yields an infrastructure cost near $0.010/kW/h ($10/MW/h) if a 25-year lifetime is assumed. It is permissible for the implementer of this function to assume that $T = \mathbf{2.19x10^5}$ hours (**25** years) if better estimates are unavailable for the lifetime of the wind farm installation.

After a wind farm exceeds its planned lifetime, a decision should be made. Thereafter, the infrastructure cost may be (a) zeroed out, (b) replaced by ongoing maintenance costs, or (c) continued as before as an ongoing replacement cost. This function should be revisited and refined when this situation will be encountered.

This function should also predict the electrical power that will be produced by the wind resource during each future interval. An explicit algorithm could be created to convert predicted weather conditions (like wind speed and direction) into electrical power output. This function will assume that experts satisfy this goal by predicting electrical power output from meteorological data that is available to them.

Block Input/Output Function Model:

Inputs:

P—Wind farm capacity/power rating.

T—Lifetime of wind farm.

IST_n—Present time series interval start times used by an example toolkit framework, where $n = \mathbf{0}, \mathbf{1}, \ldots, \mathbf{56}$. (There is no prediction to correspond with IST_n for n=**56**. This last IST is simply used to make it clear when the final interval concludes.)

Meteorological data—Predicted wind speed, wind direction, relative humidity and perhaps other weather data that experts may use to predict electrical power production for wind farms.

Outputs:

$C_{I,n}$—Time series of infrastructure cost terms expected by the Transactive Node Framework (unit: \$/h); series members correspond to IST_n. Infrastructure costs are not expected to be dynamic, but it is specified as a time series for consistency with the Transactive Node Framework.

$P_{G,n}$—Time series of predicted electrical power generated by wind farm (unit: average kW); series members correspond to IST_n.

$C_{E,n}$—Time series of energy cost terms (unit: cost per energy). Since the cost of supplying the wind energy generated is applied purely as an infrastructure cost, these energy cost terms should simply be set to zero. Note that these terms go in pair with the $P_{G,n}$ terms and are used by the Transactive Node Framework.

Pseudo Code Implementation:

1. If necessary, restate P in kW and T in h (hour).
2. Compute the infrastructure cost $C_{I,n}$ corresponding to IST_n for n, as in equation (**1**).

$$C_{I,n} = \frac{(\$2155/\text{kW}) \times P}{T}, \text{for } n = 0,1,\ldots,55 \qquad \textbf{(1)}$$

3. Predict the average wind electrical power output $P_{G,n}$ that will be generated during each future interval corresponding to IST_n for n.
4. Output $C_{E,n} = 0$, for $n = 0,1,\ldots,55$.

6.3.8 Incentive function – Solar energy (Function 2.2)

Description:

This function addresses solar power generation and is to be applied at transactive nodes which have and represent solar farm energy that is produced within or near their electrical boundaries to encourage the use of solar energy when and near where it is generated. This function is applicable to energy produced by a solar farm or may be applied to aggregated output from multiple solar farms.

The cost of supplying the solar energy generated is applied as an infrastructure cost, in units of cost per time, consistent with the Transactive Node Framework. For simplicity, the infrastructure cost will use the $7.5/W capacity-weighted average installed cost for a solar farm. The infrastructure cost of a solar farm can thus be estimated if its capacity is known. This cost shall then be spread over the lifetime T of the solar farm.

Note that this calculation typically yields an infrastructure cost near $0.034/kW/h ($34/MW/h) if a 25-year lifetime is assumed. It is permissible for the implementer of this function to assume that $T = 2.19\text{x}10^5$ hours (25 years) if better estimates are unavailable for the lifetime of the solar farm installation.

After a solar farm exceeds its planned lifetime, a decision should be made. Thereafter, the infrastructure cost may be (a) zeroed out, (b) replaced by ongoing maintenance costs, or (c) continued as before as an ongoing replacement cost. This function should be revisited and refined when this situation will be encountered.

This function should also predict the electrical power that will be produced by the solar resource during each future interval. An explicit algorithm could be created to convert predicted weather conditions (like solar irradiance and temperature) into electrical power output. This function will assume that experts satisfy this goal by predicting electrical power output from meteorological data that is available to them.

Block Input/Output Function Model:

Inputs:

P—Solar farm capacity/power rating.

T—Lifetime of solar farm.

IST_n—Present time series interval start times used by the toolkit framework, where $n = 0, 1, \ldots, 56$. (There is no prediction to correspond with IST_n for n=56. This last IST is simply used to make it clear when the final interval concludes.)

Meteorological data—Solar irradiance, temperature, and perhaps other weather data that experts may use to predict electrical power production for solar farms.

Outputs:

DEMANDE OU BREVET VOLUMINEUX

LA PRÉSENTE PARTIE DE CETTE DEMANDE OU CE BREVET COMPREND PLUS D'UN TOME.

CECI EST LE TOME 1 DE 2
CONTENANT LES PAGES 1 À 368

NOTE : Pour les tomes additionels, veuillez contacter le Bureau canadien des brevets

JUMBO APPLICATIONS/PATENTS

THIS SECTION OF THE APPLICATION/PATENT CONTAINS MORE THAN ONE VOLUME

THIS IS VOLUME 1 OF 2
CONTAINING PAGES 1 TO 368

NOTE: For additional volumes, please contact the Canadian Patent Office

NOM DU FICHIER / FILE NAME :

NOTE POUR LE TOME / VOLUME NOTE:

CLAIMS:

1) A method for operating a transactive node in a market-based electrical-energy-allocation system, comprising:

by computing hardware:

computing incentive signal data, the incentive signal data comprising data indicative of a cost of electric energy at the transactive node at a current time interval and data indicative of a forecasted cost of electric energy at the transactive node at one or more future time intervals, wherein the incentive signal data further comprises data indicative of a confidence level that the data indicative of the cost of electric energy at the transactive node at the current time interval is accurate or data indicating a confidence level that the data indicative of the forecasted cost of electric energy at the transactive node at the one or more future time intervals is accurate;

computing feedback signal data, the feedback signal data comprising data indicative of an electric load at the transactive node at the current time interval and data indicative of a forecasted load for electric energy at the transactive node at the one or more future time intervals; and

transmitting the incentive signal data and the feedback signal data.

2) The method of claim **1)**, wherein the data indicative of the cost of electric energy comprises data indicative of a cost of real electrical energy, reactive electrical energy, or a combination of both real and reactive electrical energies at the transactive node at the current time interval, and wherein the data indicative of the forecasted cost of electric energy comprises data indicative of a forecasted cost of real electrical energy, reactive electrical energy, or a combination of both real and reactive electrical energies at the transactive node at the one or more future time intervals.

3) The method of claim **1)**, wherein the data indicative of the electric load comprises data indicative of a real electrical load, reactive electrical load, or a combination of both real and reactive electrical loads at the transactive node at the current time interval, and wherein the data indicative of the forecasted load for electric energy comprises data indicative of a forecasted load of real electrical load, reactive electrical load, or a combination of both real and reactive electrical loads at the transactive node at the one or more future time intervals.

4) The method of claim **1)**, wherein the incentive signal data further comprises data indicating a confidence level that the data indicative of the cost of electric energy at the transactive node at the current time interval is accurate, and data indicating a confidence level that the data indicative of the forecasted cost of electric energy at the transactive node at the one or more future time intervals is accurate.

5) The method of claim **1)**, wherein the feedback signal data further comprises data indicating a confidence level that the data indicative of the electric load at the transactive node at the current time interval is accurate, and data indicating a confidence level that the data indicative of the forecasted load for electric energy at the transactive node at the one or more future time intervals is accurate.

6) The method of claim **1)**, wherein the method further comprises receiving incentive signal data and feedback signal data from one or more neighboring transactive nodes, wherein the computing the incentive signal data is based at least in part on the received incentive signal data, and wherein the computing the feedback signal data is based at least in part on the received feedback signal data.

7) One or more non-transitory computer-readable media storing computer-readable instructions for causing computer to perform the method of claim **1)**.

8) A transactive node comprising computing hardware configured to perform the method of claim **1)**.

9) A method for operating a transactive node in a market-based electrical-energy-allocation system, comprising:

by computing hardware:

receiving incentive signal data at the transactive node from two or more neighboring transactive nodes, the incentive signal data from the two or more neighboring transactive nodes comprising data indicative of at least a cost of electric energy at a current time interval;

computing aggregated incentive signal data based at least in part on the incentive signal data from the two or more neighboring transactive nodes; and

transmitting the aggregated incentive signal data to a further transactive node;

wherein the received incentive signal data further includes data indicating a confidence level of the received incentive signal data, or wherein the transmitted incentive signal data further includes data indicating a confidence level of the transmitted incentive signal data.

10) The method of claim **9)**, wherein the received incentive signal data and the transmitted aggregated incentive signal data comprise data indicative of a cost of real electrical energy, reactive electrical energy, or a combination of both real and reactive electrical energies.

11) The method of claim **9)**, wherein the aggregated incentive signal data comprises a weighted sum of the incentive signal data from the two or more neighboring transactive nodes.

12) The method of claim **9)**, wherein the aggregated incentive signal data is further modified to provide an incentive or disincentive to the further transactive node based on local conditions at the transactive node.

13) The method of claim **9)**, wherein the received incentive signal data comprises data indicative of the cost of electric energy at the current time interval and data indicative of a forecasted cost of electric energy at one or more future time intervals, and wherein the aggregated incentive signal data comprises data indicative of the aggregated cost of electric energy at the current time interval and data indicative of a forecasted aggregated cost of electric energy at one or more future time intervals.

14) The method of claim **9)**, wherein the method further comprises:

receiving feedback signal data at the transactive node from the two or more neighboring transactive nodes, the feedback signal data from the two or more neighboring transactive nodes comprising data indicative of at least an electric load for electric energy at a current time interval;

computing aggregated feedback signal data based at least in part on the feedback signal data from the two or more neighboring transactive nodes; and

transmitting the aggregated feedback signal data to the further transactive node.

15) The method of claim **14)**, wherein the received feedback signal data comprises data indicative of the electric load for electric energy at the current time interval and data indicative of a forecasted load of electric energy at the one or more future time intervals, and wherein the aggregated feedback signal data comprises data indicative of the aggregated load of electric energy at the current time interval and data indicative of a forecasted aggregated load of electric energy at one or more future time intervals.

16) One or more non-transitory computer-readable media storing computer-readable instructions for causing computer to perform the method of claim **9)**.

17) A transactive node comprising computing hardware configured to perform the method of claim **9)**.

18) A method for operating a transactive node in a market-based electrical-energy-allocation system, comprising:

by computing hardware:

receiving feedback signal data at a transactive node from two or more neighboring transactive nodes, the feedback signal data from the two or more neighboring transactive nodes comprising data indicative of at least an electric load for electric energy at a current time interval;

computing aggregated feedback signal data based at least in part on the feedback signal data from the two or more neighboring transactive nodes; and

transmitting the aggregated feedback signal data to a further transactive node;

wherein the received feedback signal data further includes data indicating a confidence level of the received feedback signal data, or

wherein the transmitted feedback signal data further includes data indicating a confidence level of the transmitted feedback signal data.

19) The method of claim **18)**, wherein the received feedback signal data and the transmitted aggregated feedback signal data comprise data indicative of a real electrical load, reactive electrical load, or a combination of both real and reactive electrical loads.

20) The method of claim **18)**, wherein the received feedback signal data comprises data indicative of the electric load of electric energy at the current time interval and data indicative of a forecasted load of electric energy at one or more future time intervals, and wherein the aggregated feedback signal data comprises data indicative of the aggregated load of electric energy at the current time interval and data indicative of a forecasted aggregated load of electric energy at the one or more future time intervals.

21) The method of claim **18)**, wherein the method further comprises:

receiving incentive signal data at the transactive node from the two or more neighboring transactive nodes, the incentive signal data from the two or more neighboring transactive nodes comprising data indicative of at least a cost of electric energy at the current time interval;

computing aggregated incentive signal data based at least in part on the incentive signal data from the two or more neighboring transactive nodes; and

transmitting the aggregated incentive signal data to the further transactive node.

22) The method of claim **21)**, wherein the received incentive signal data comprises data indicative of the cost of electric energy at the current time interval and data indicative of a forecasted cost of electric energy at the one or more future time intervals, and wherein the aggregated incentive signal data comprises data indicative of the aggregated cost of electric energy at the current time interval and data indicative of a forecasted aggregated cost of electric energy at one or more future time intervals.

23) One or more non-transitory computer-readable media storing computer-readable instructions for causing computer to perform the method of claim **18)**.

24) A transactive node comprising computing hardware configured to perform the method of claim **18)**.

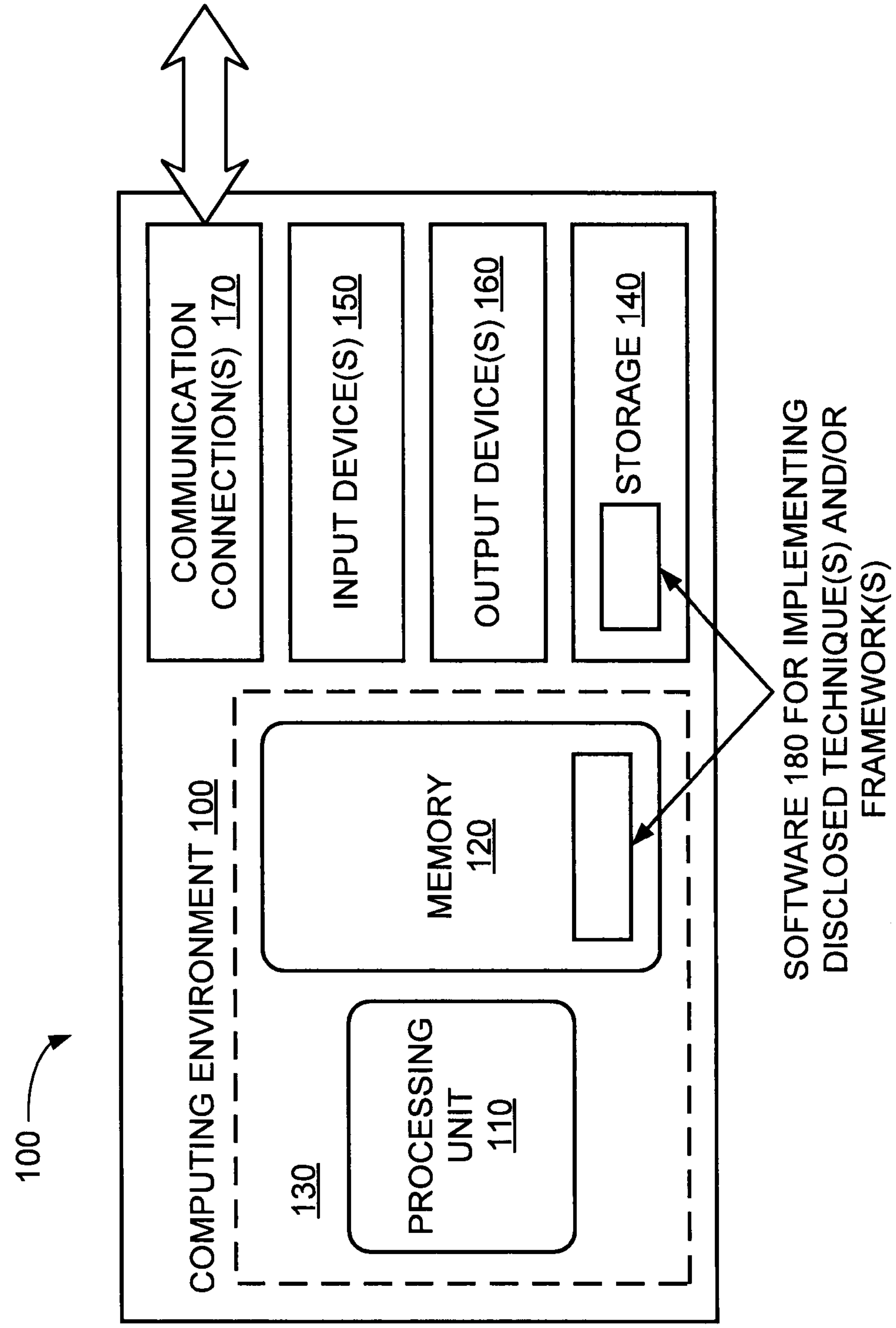

100
COMPUTING ENVIRONMENT 100
130
PROCESSING UNIT 110
MEMORY 120
COMMUNICATION CONNECTION(S) 170
INPUT DEVICE(S) 150
OUTPUT DEVICE(S) 160
STORAGE 140
SOFTWARE 180 FOR IMPLEMENTING DISCLOSED TECHNIQUE(S) AND/OR FRAMEWORK(S)

FIG. 1

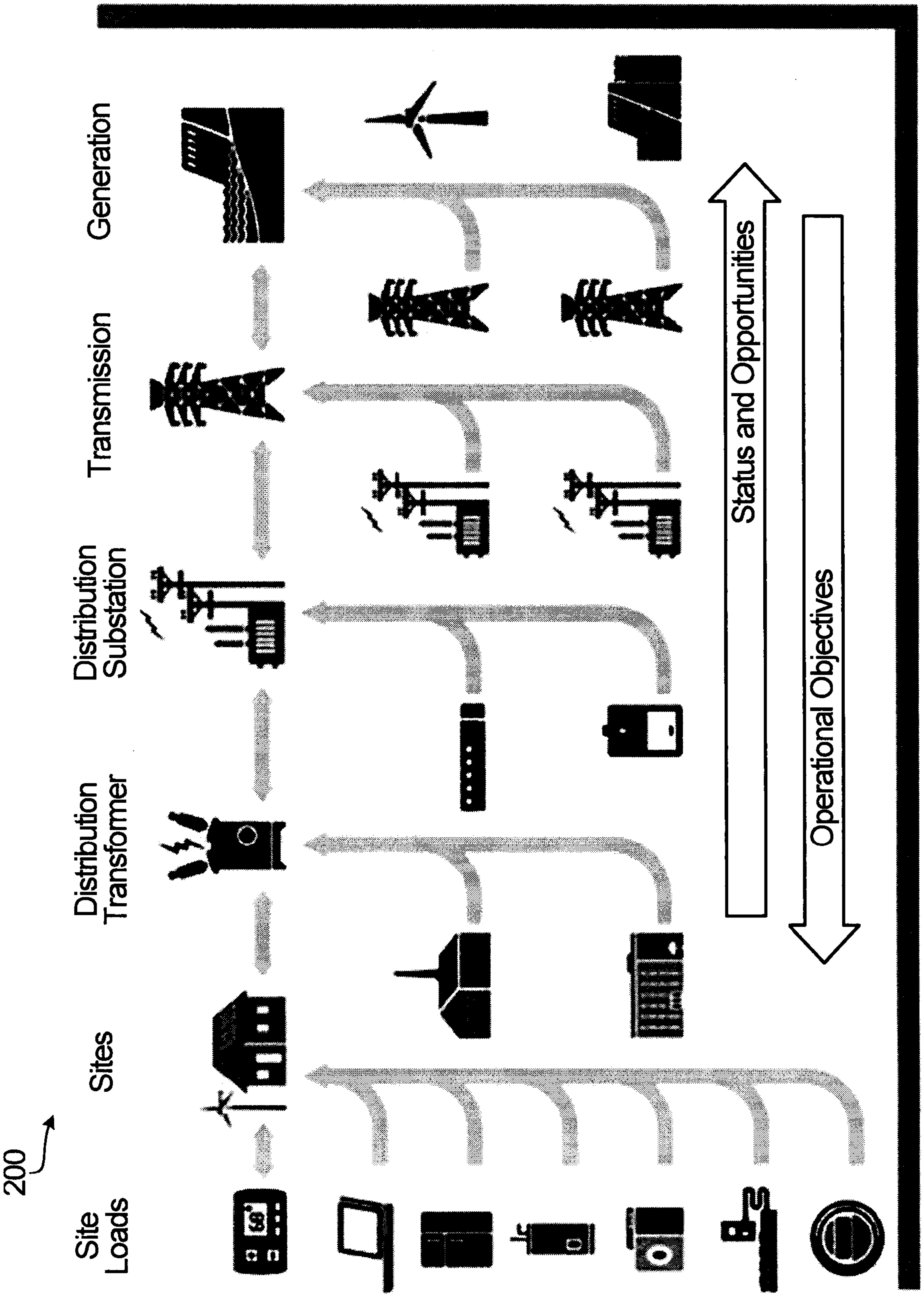

200
Site Loads
Sites
Distribution Transformer
Distribution Substation
Transmission
Generation
Status and Opportunities
Operational Objectives

FIG. 2

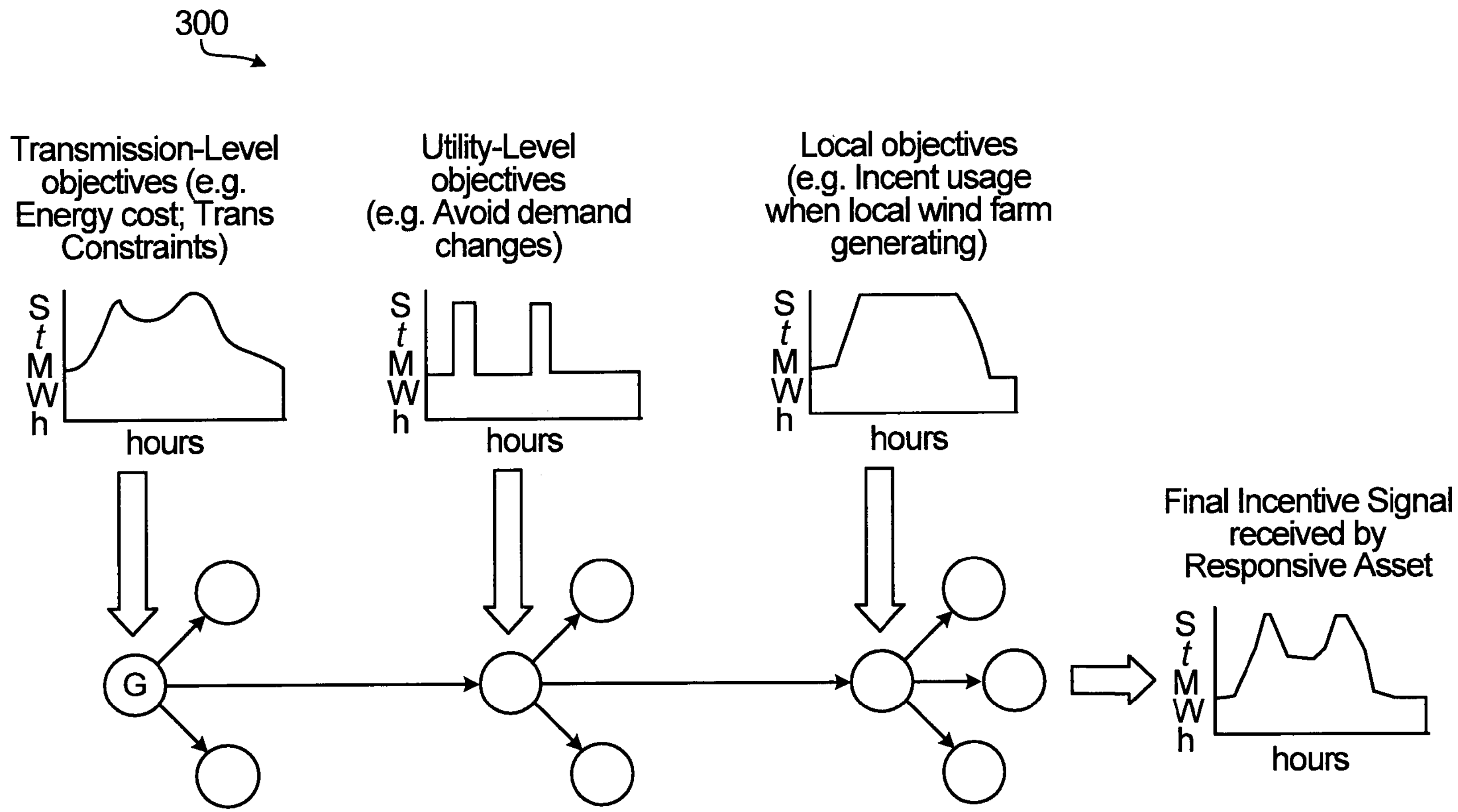

300
Transmission-Level objectives (e.g. Energy cost; Trans Constraints)
St MWh
hours
Utility-Level objectives (e.g. Avoid demand changes)
St MWh
hours
Local objectives (e.g. Incent usage when local wind farm generating)
St MWh
hours
G
Final Incentive Signal received by Responsive Asset
St MWh
hours

FIG. 3

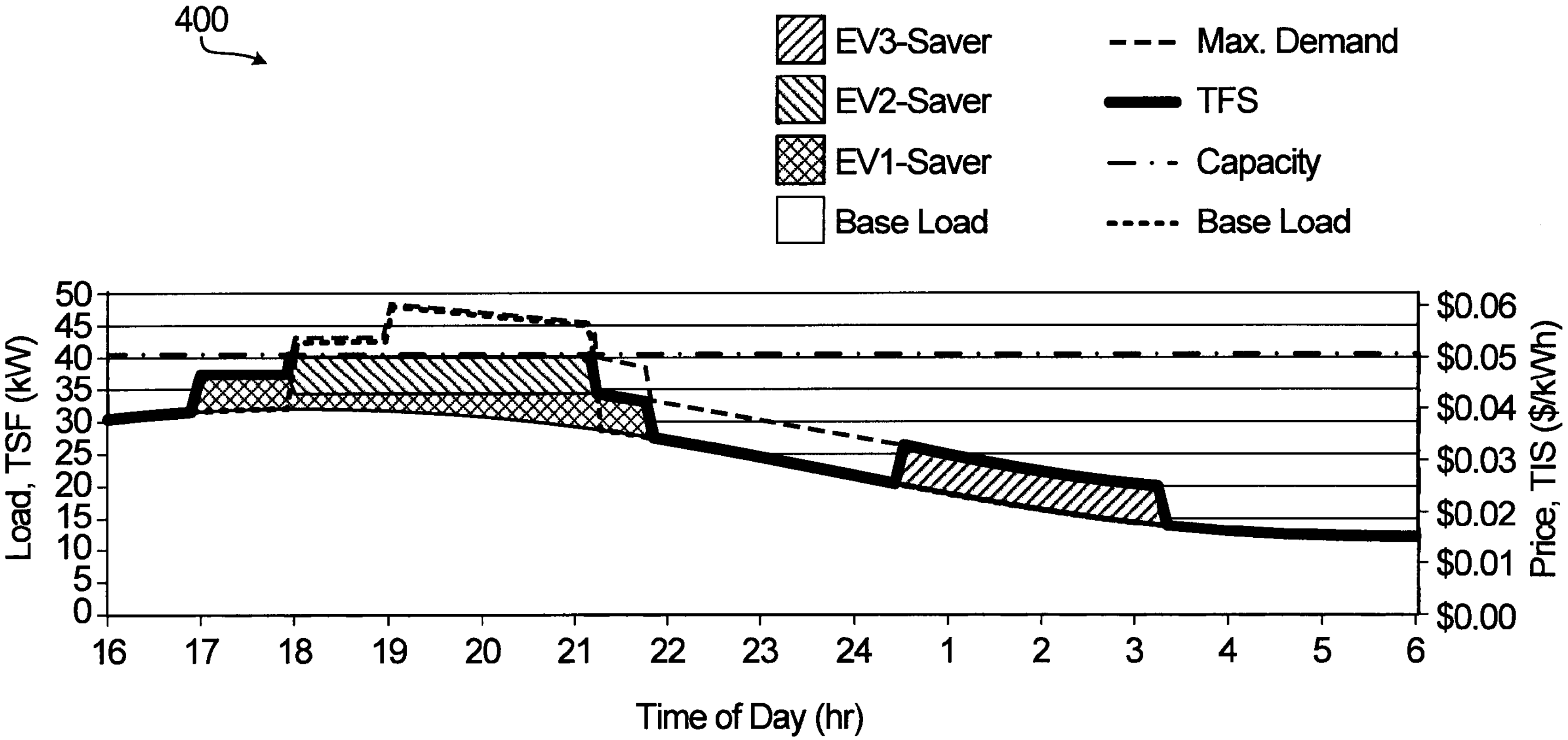

400
EV3-Saver
EV2-Saver
EV1-Saver
Base Load
Max. Demand
TFS
Capacity
Base Load
Load, TSF (kW)
50
45
40
35
30
25
20
15
10
5
0
Price, TIS ($/kWh)
$0.06
$0.05
$0.04
$0.03
$0.02
$0.01
$0.00
16
17
18
19
20
21
22
23
24
1
2
3
4
5
6
Time of Day (hr)

FIG. 4

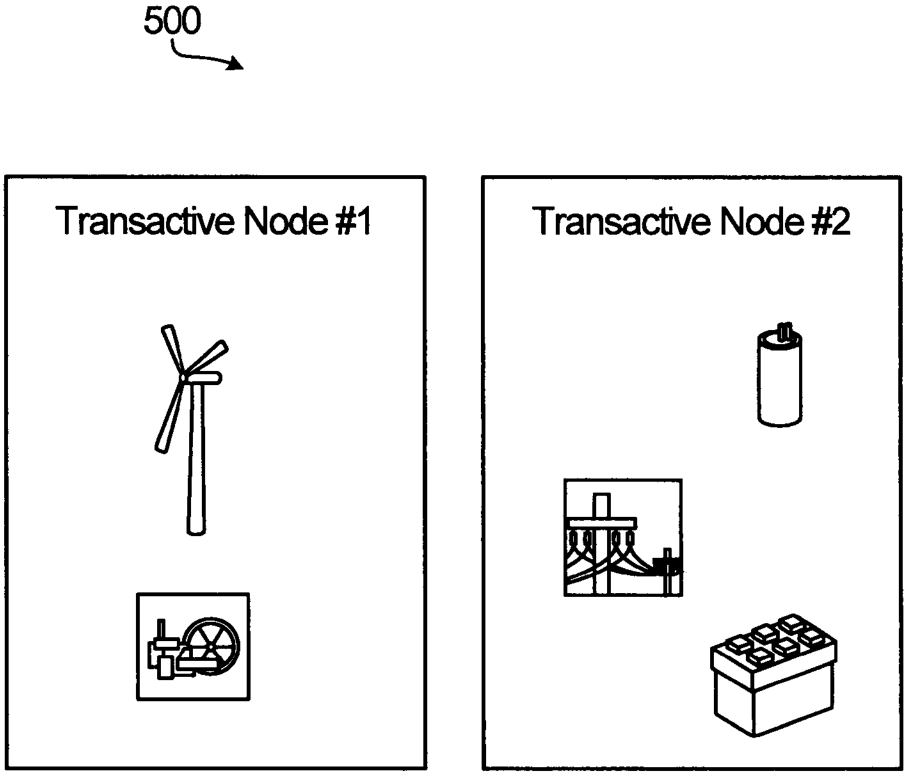


FIG. 5

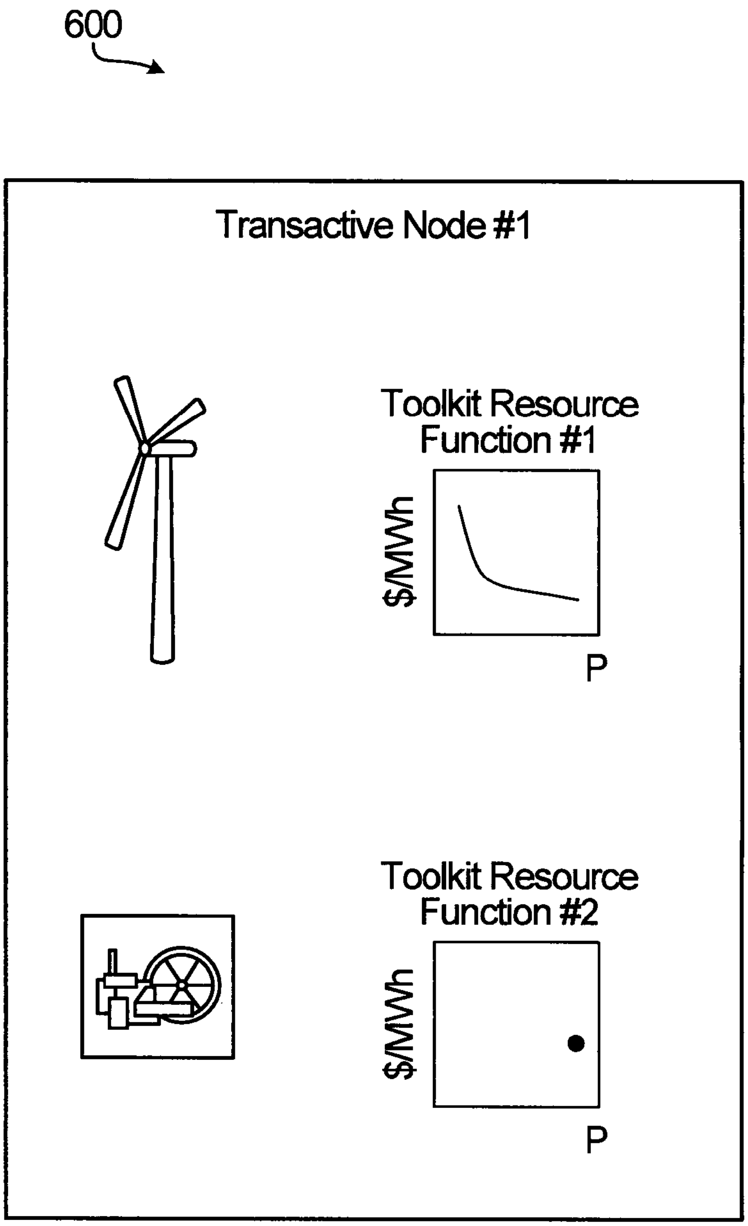

600
Transactive Node #1
Toolkit Resource Function #1
$/MWh
P
Toolkit Resource Function #2
$/MWh
P

FIG. 6

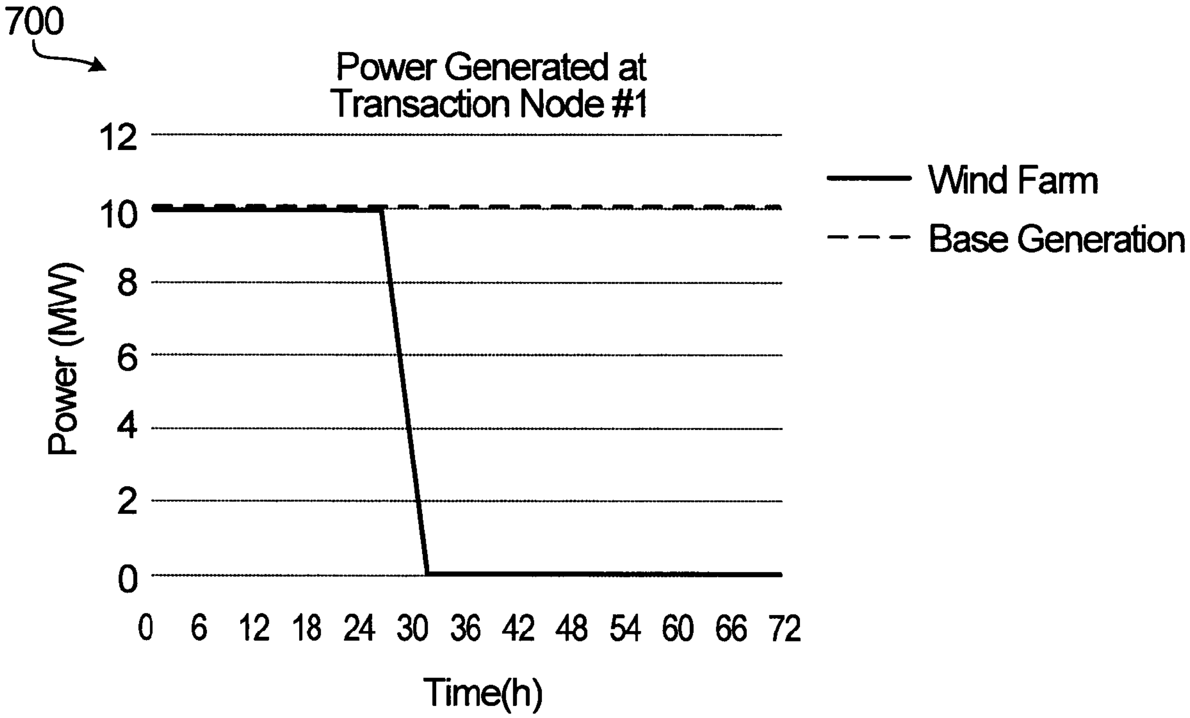

700
Power Generated at Transaction Node #1
12
10
8
6
4
2
0
Power (MW)
0
6
12
18
24
30
36
42
48
54
60
66
72
Time(h)
Wind Farm
Base Generation

FIG. 7

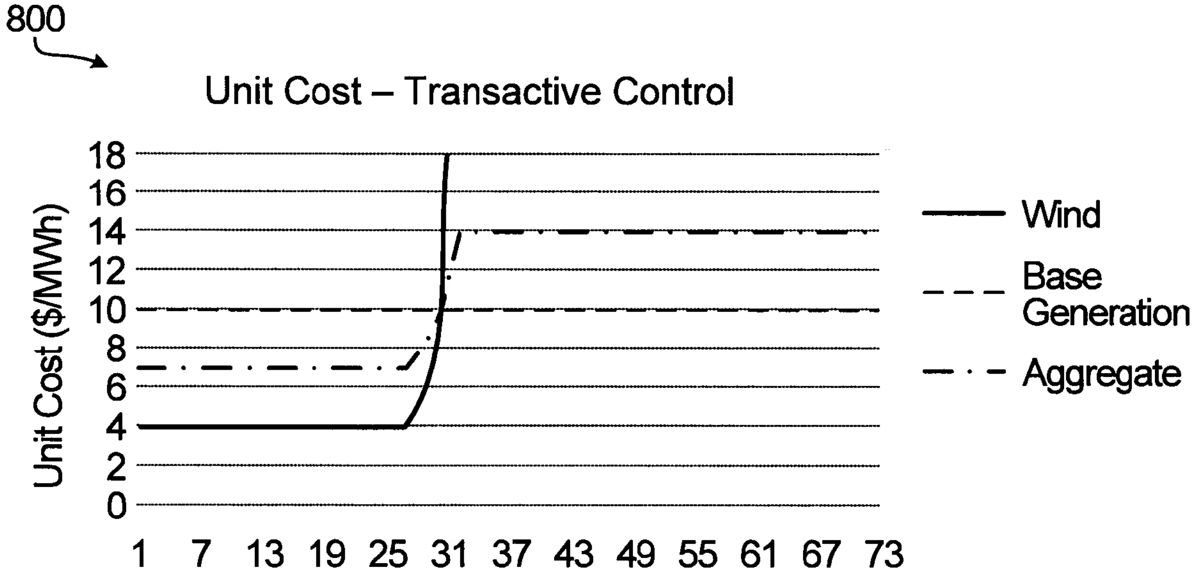

800
Unit Cost – Transactive Control
18
16
14
12
10
8
6
4
2
0
Unit Cost ($/MWh)
1
7
13
19
25
31
37
43
49
55
61
67
73
Wind
Base Generation
Aggregate

FIG. 8

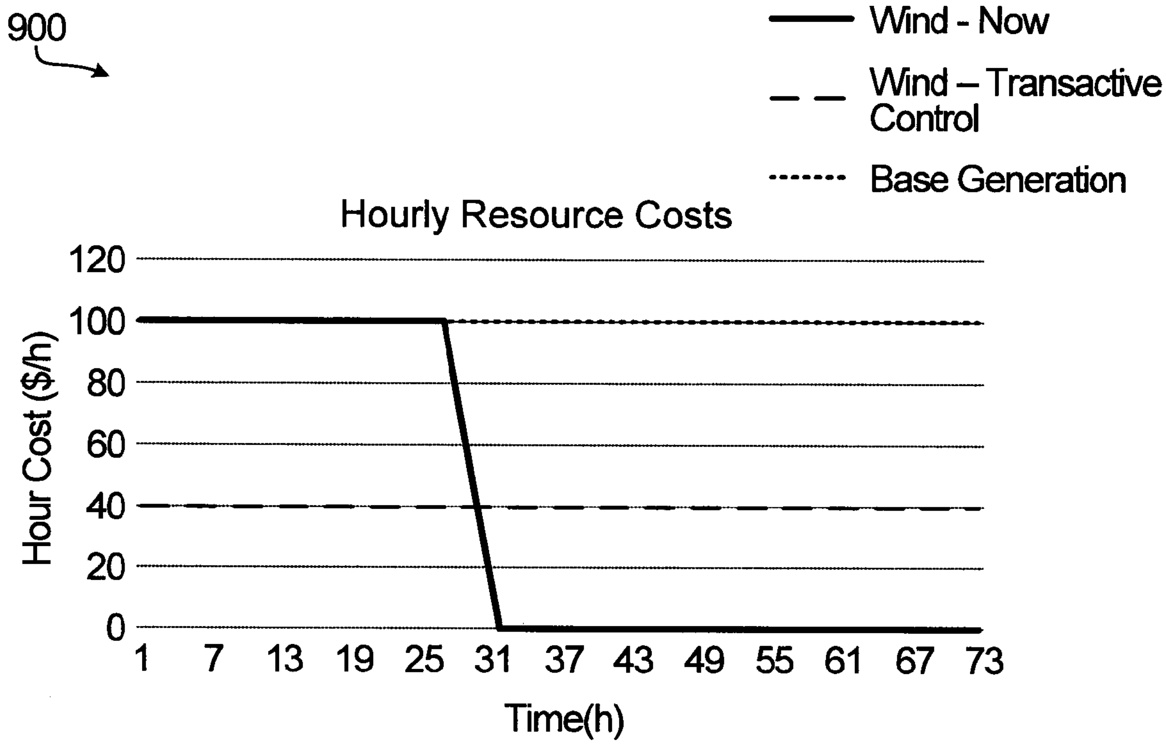

900
Wind - Now
Wind – Transactive Control
Base Generation
Hourly Resource Costs
120
100
80
60
40
20
0
Hour Cost ($/h)
1
7
13
19
25
31
37
43
49
55
61
67
73
Time(h)

FIG. 9

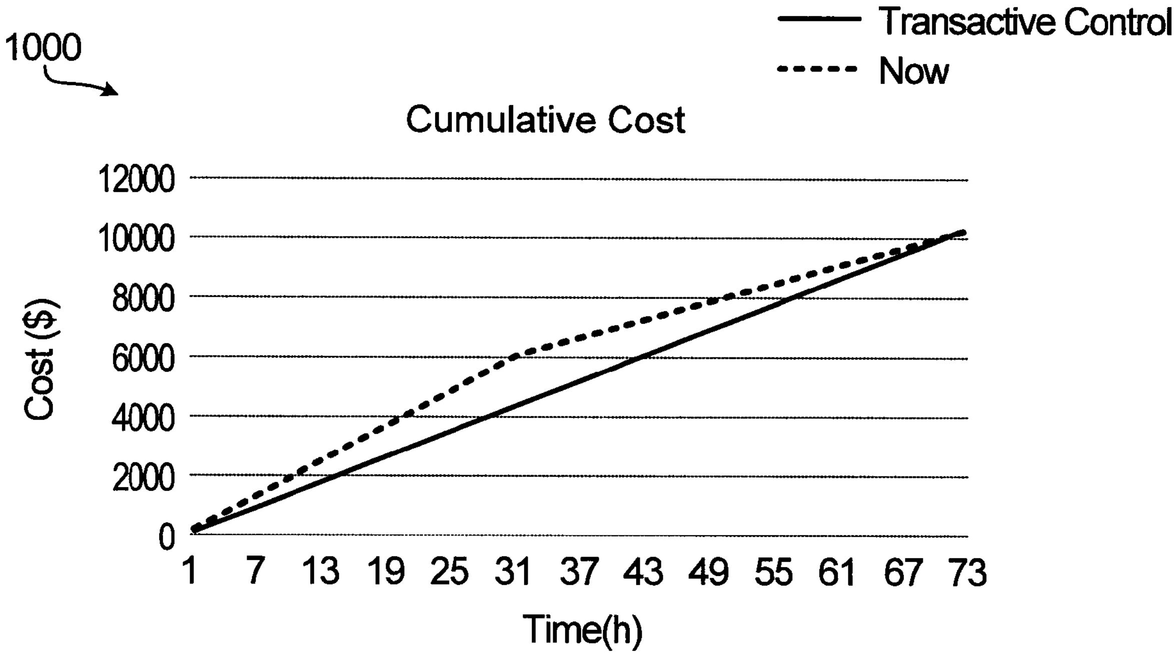

1000
Transactive Control
Now
Cumulative Cost
12000
10000
8000
6000
4000
2000
0
Cost ($)
1
7
13
19
25
31
37
43
49
55
61
67
73
Time(h)

FIG. 10

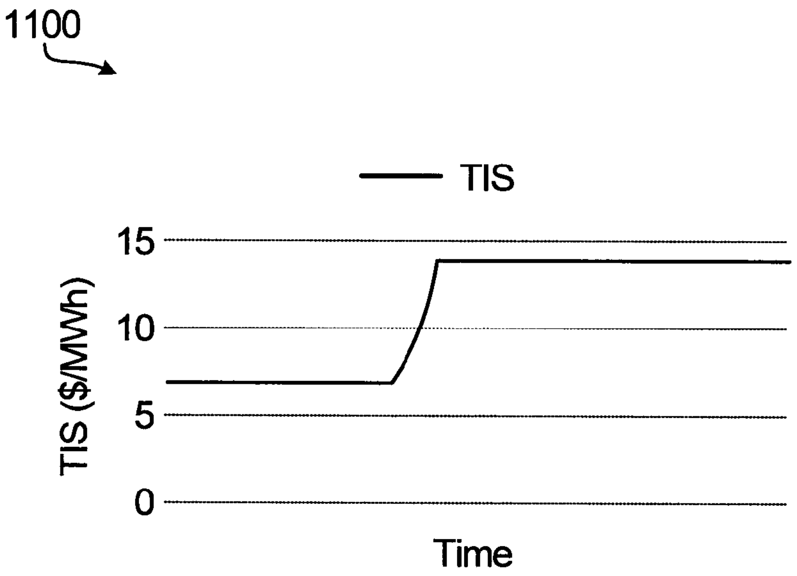


FIG. 11

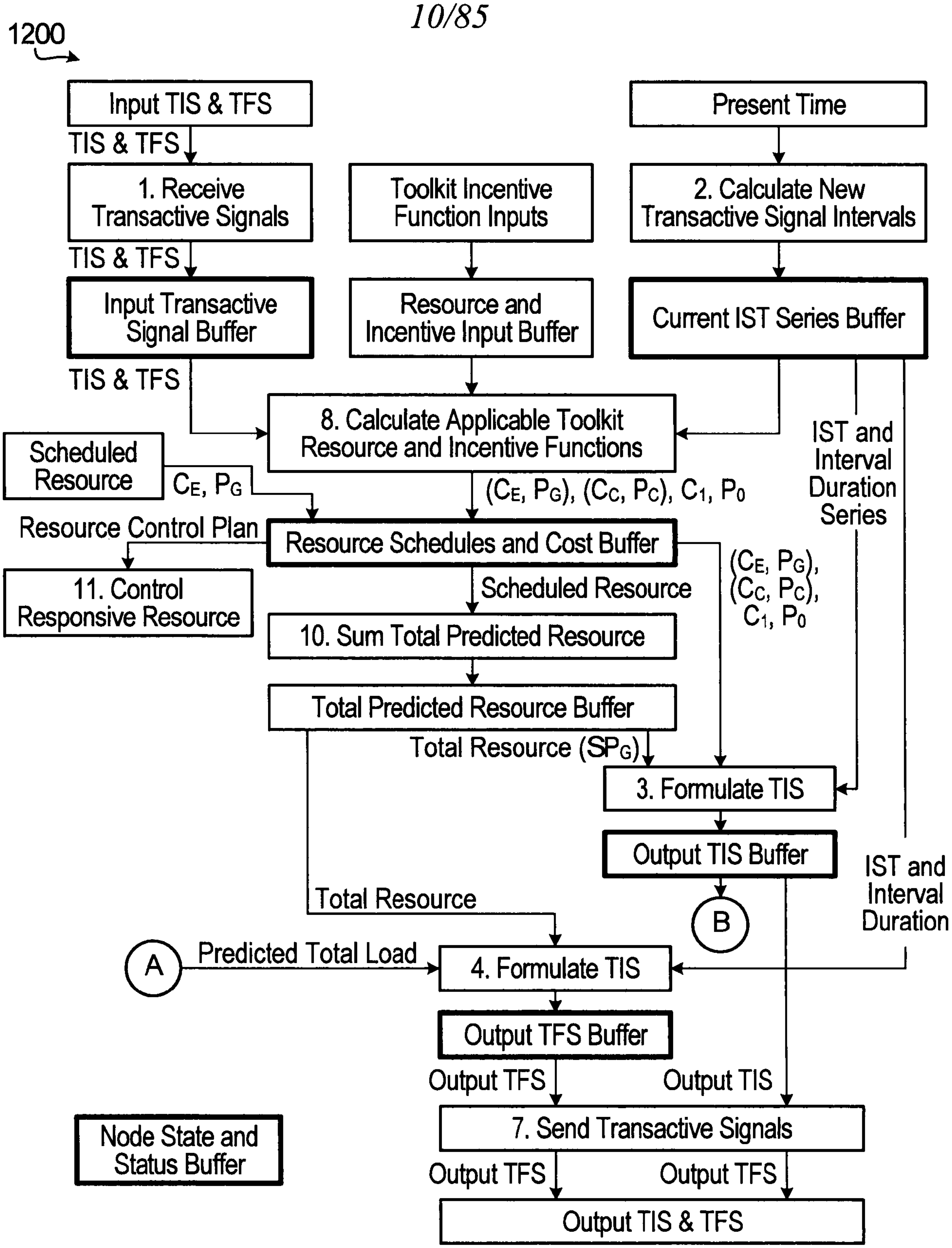

1200
Input TIS & TFS
TIS & TFS
1. Receive Transactive Signals
TIS & TFS
Input Transactive Signal Buffer
TIS & TFS
Toolkit Incentive Function Inputs
Resource and Incentive Input Buffer
Present Time
2. Calculate New Transactive Signal Intervals
Current IST Series Buffer
8. Calculate Applicable Toolkit Resource and Incentive Functions
IST and Interval Duration Series
Scheduled Resource
C_E, P_G
(C_E, P_G), (C_C, P_C), C_1, P_0
Resource Control Plan
Resource Schedules and Cost Buffer
11. Control Responsive Resource
Scheduled Resource
(C_E, P_G), (C_C, P_C), C_1, P_0
10. Sum Total Predicted Resource
Total Predicted Resource Buffer
Total Resource (SP_G)
3. Formulate TIS
Output TIS Buffer
B
IST and Interval Duration
Total Resource
A
Predicted Total Load
4. Formulate TIS
Output TFS Buffer
Output TFS
Output TIS
7. Send Transactive Signals
Output TFS
Output TFS
Output TIS & TFS
Node State and Status Buffer

FIG. 12A

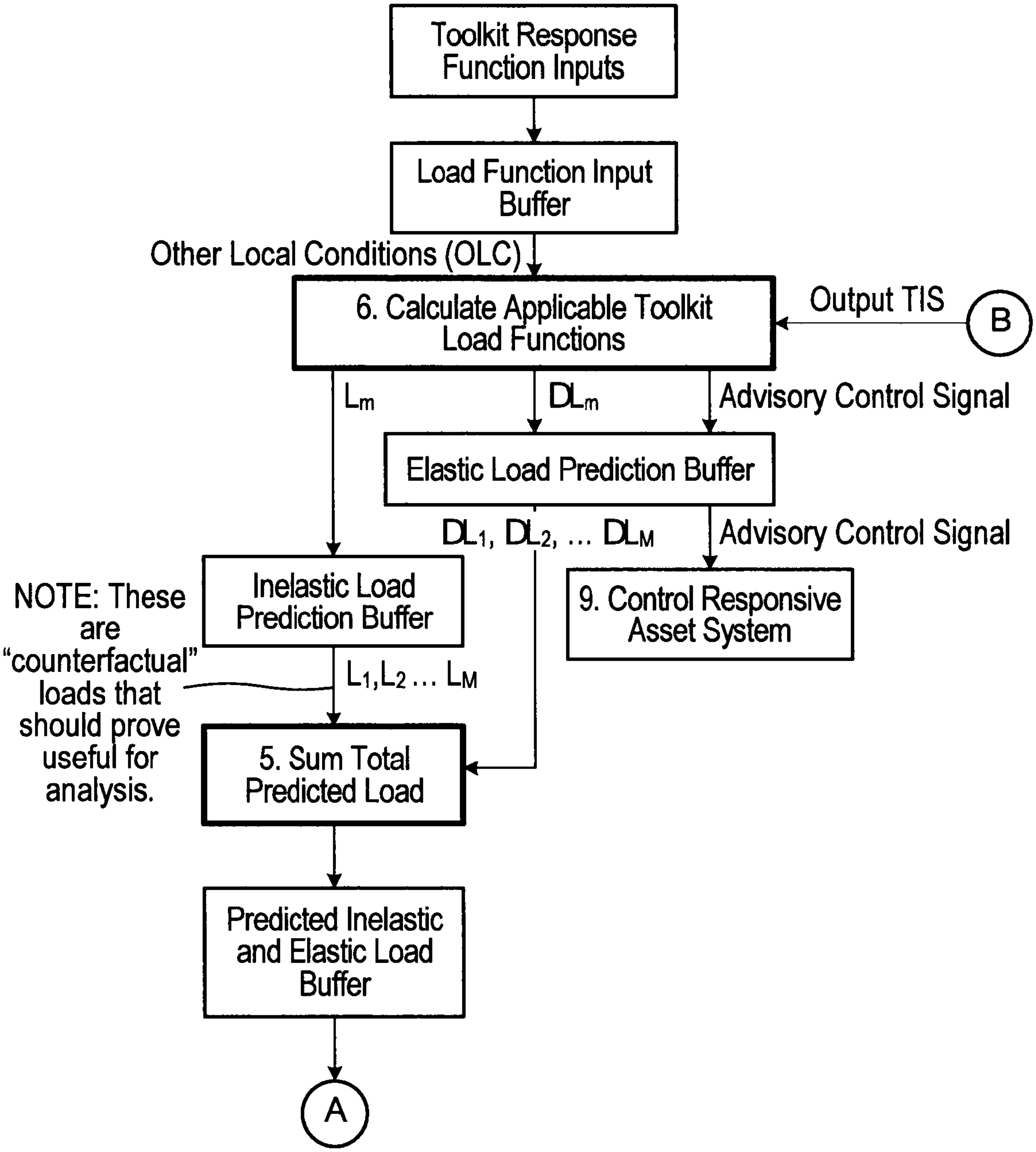

Toolkit Response Function Inputs
Load Function Input Buffer
Other Local Conditions (OLC)
6. Calculate Applicable Toolkit Load Functions
Output TIS
B
Lm
DLm
Advisory Control Signal
Elastic Load Prediction Buffer
DL1, DL2, ... DLM
Advisory Control Signal
9. Control Responsive Asset System
Inelastic Load Prediction Buffer
NOTE: These are "counterfactual" loads that should prove useful for analysis.
L1,L2 ... LM
5. Sum Total Predicted Load
Predicted Inelastic and Elastic Load Buffer
A

FIG. 12B

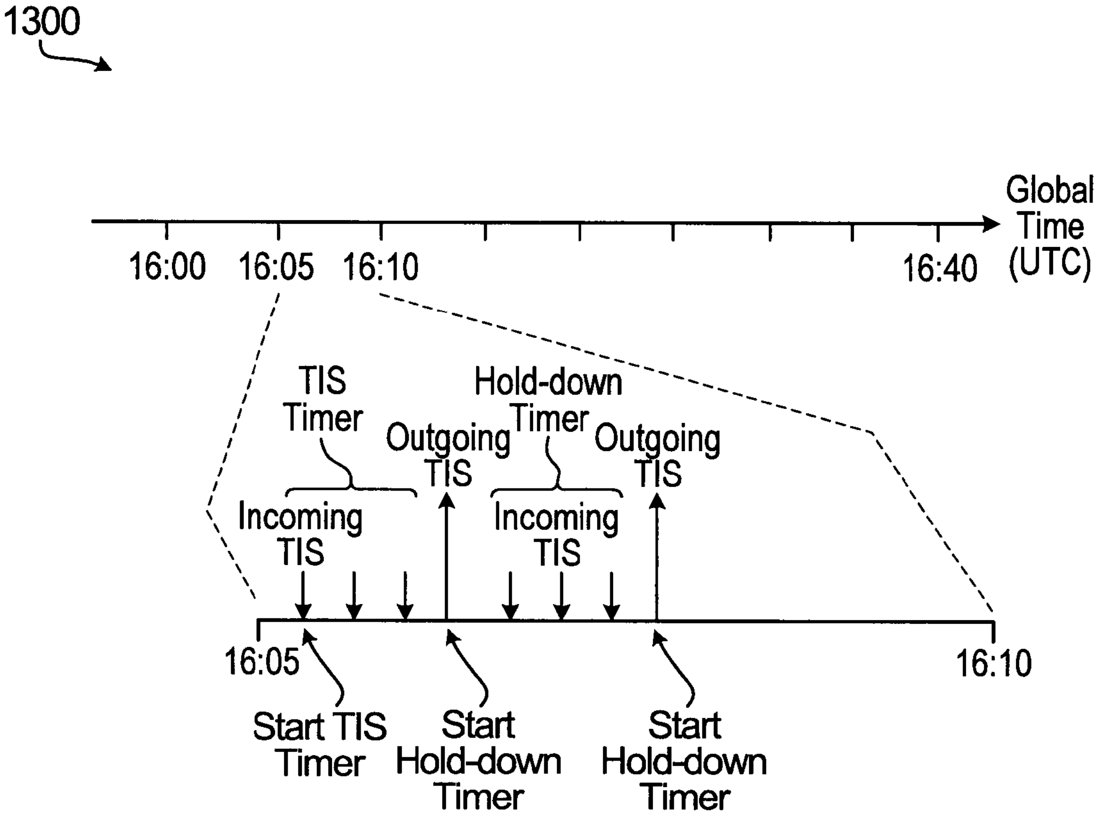

1300
Global Time (UTC)
16:00
16:05
16:10
16:40
TIS Timer
Hold-down Timer
Outgoing TIS
Outgoing TIS
Incoming TIS
Incoming TIS
16:05
16:10
Start TIS Timer
Start Hold-down Timer
Start Hold-down Timer

FIG. 13

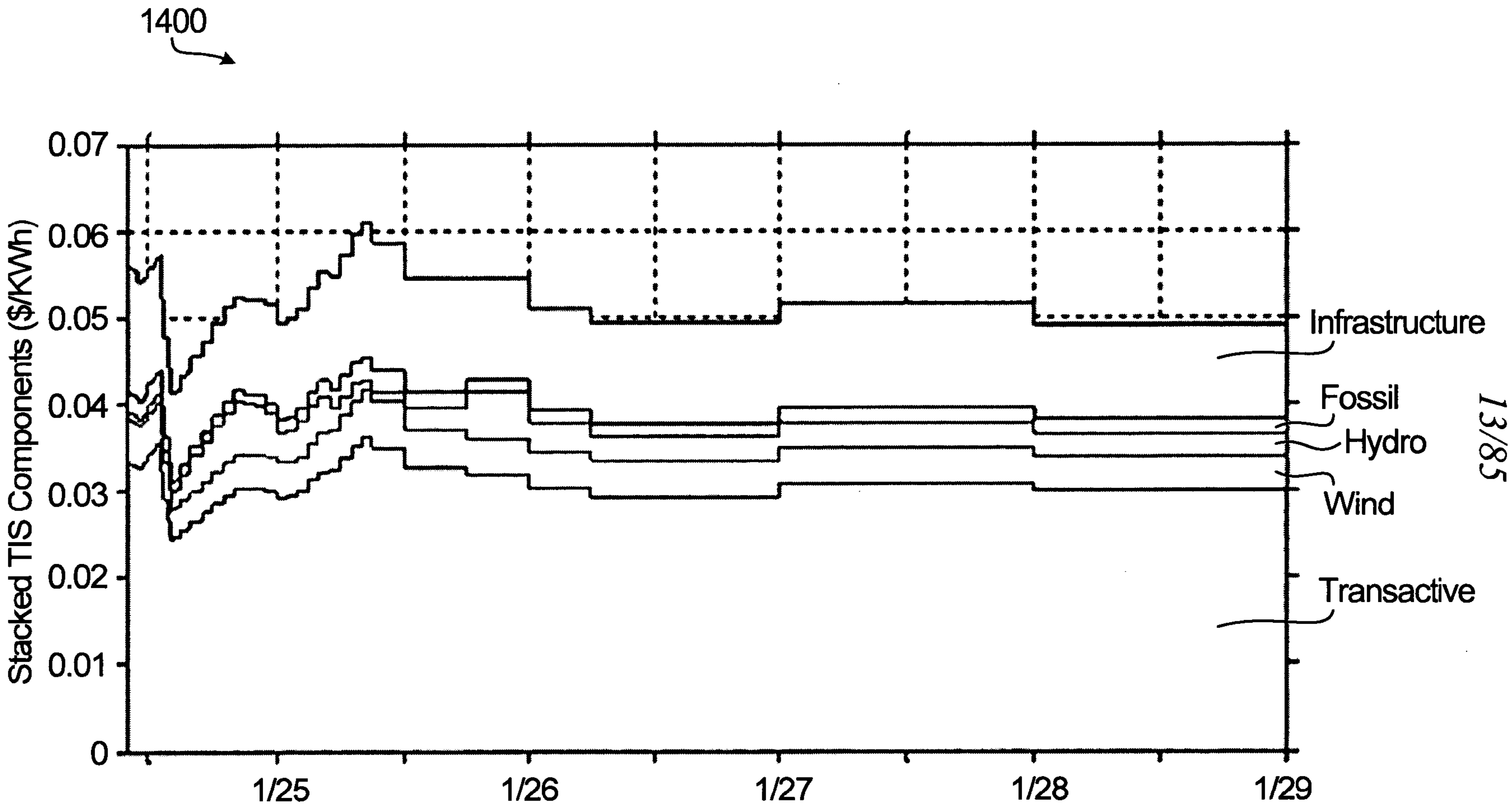
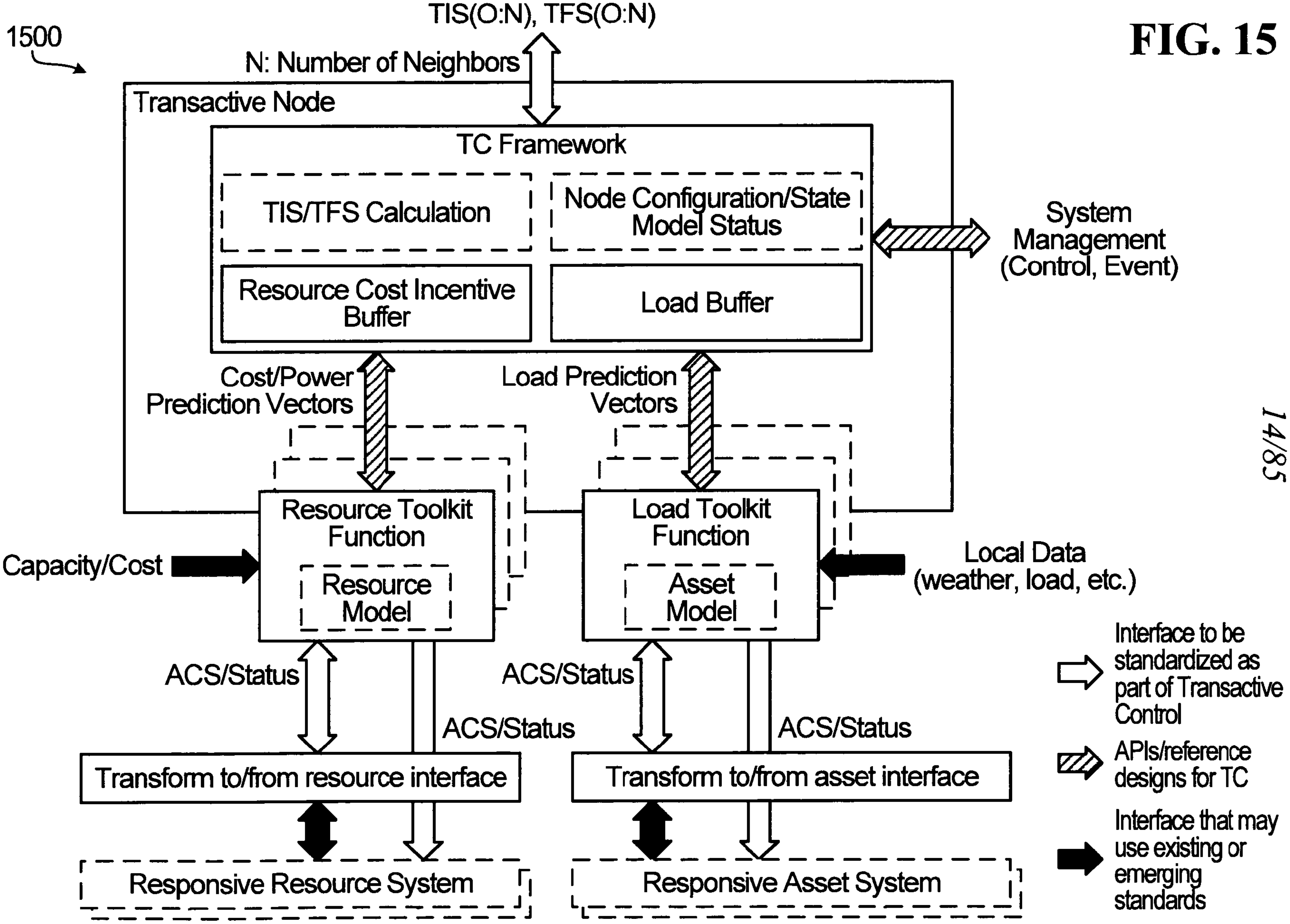

1400
Stacked TIS Components ($/KWh)
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
1/25
1/26
1/27
1/28
1/29
Infrastructure
Fossil
Hydro
Wind
Transactive

FIG. 14

FIG. 15


1500
TIS(O:N), TFS(O:N)
N: Number of Neighbors
Transactive Node
TC Framework
TIS/TFS Calculation
Node Configuration/State Model Status
Resource Cost Incentive Buffer
Load Buffer
System Management (Control, Event)
Cost/Power Prediction Vectors
Load Prediction Vectors
Resource Toolkit Function
Resource Model
Load Toolkit Function
Asset Model
Capacity/Cost
Local Data (weather, load, etc.)
ACS/Status
ACS/Status
ACS/Status
ACS/Status
Transform to/from resource interface
Transform to/from asset interface
Responsive Resource System
Responsive Asset System
Interface to be standardized as part of Transactive Control
APIs/reference designs for TC
Interface that may use existing or emerging standards

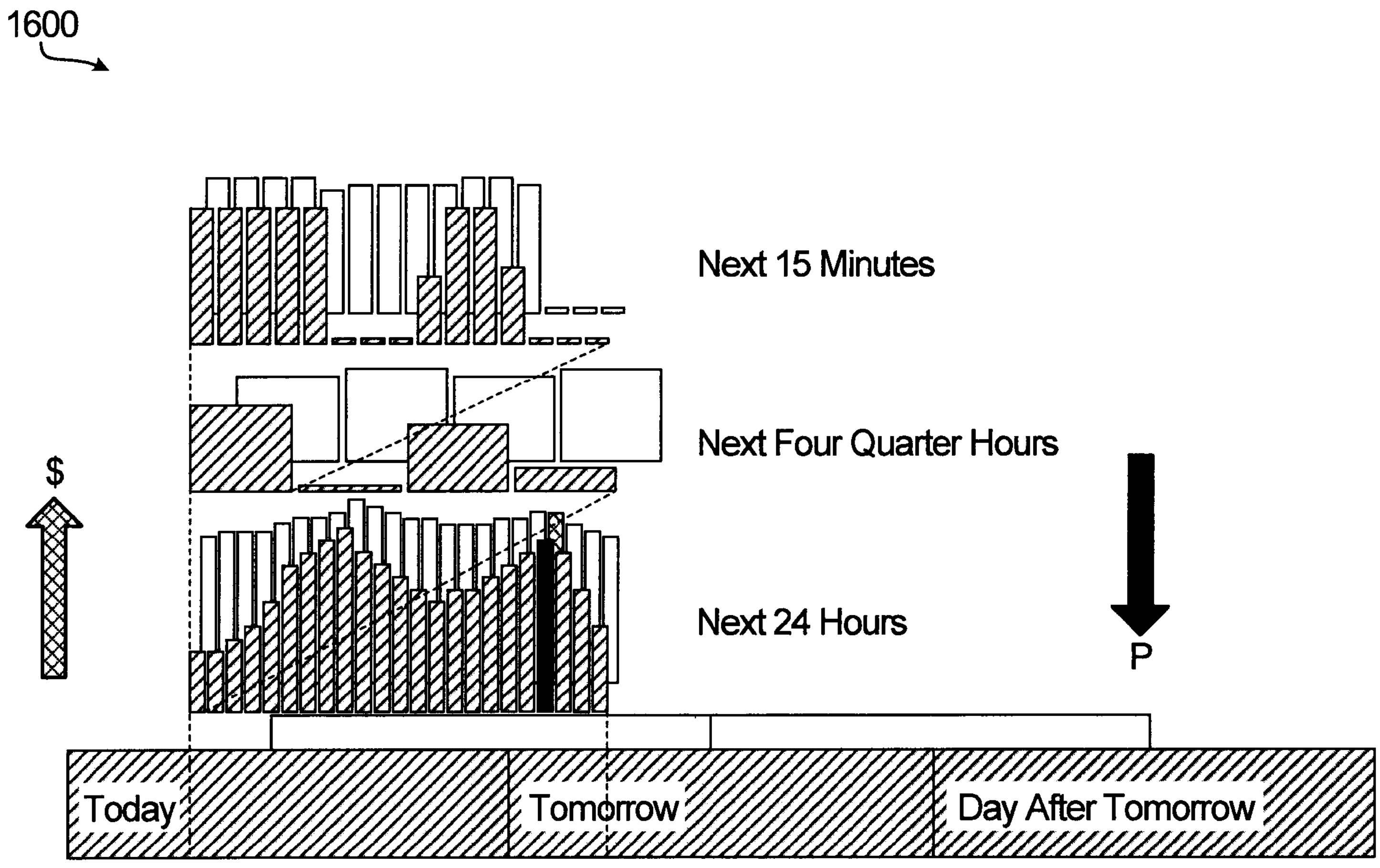

1600
$
Next 15 Minutes
Next Four Quarter Hours
Next 24 Hours
P
Today
Tomorrow
Day After Tomorrow

FIG. 16

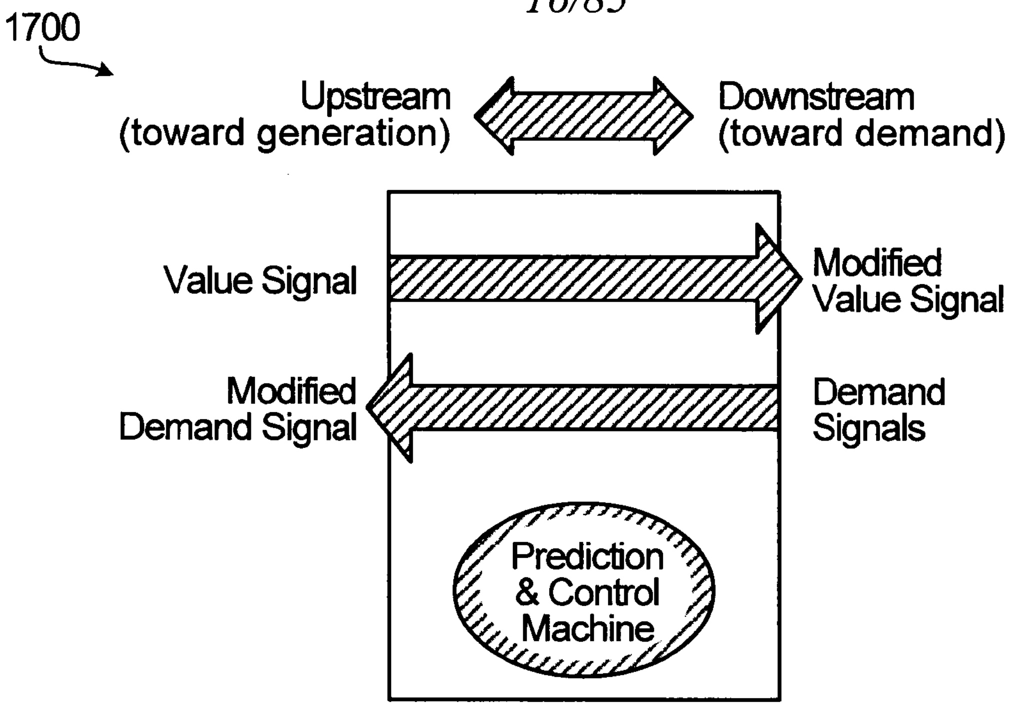

1700
Upstream
(toward generation)
Downstream
(toward demand)
Value Signal
Modified
Value Signal
Modified
Demand Signal
Demand
Signals
Prediction
& Control
Machine

FIG. 17

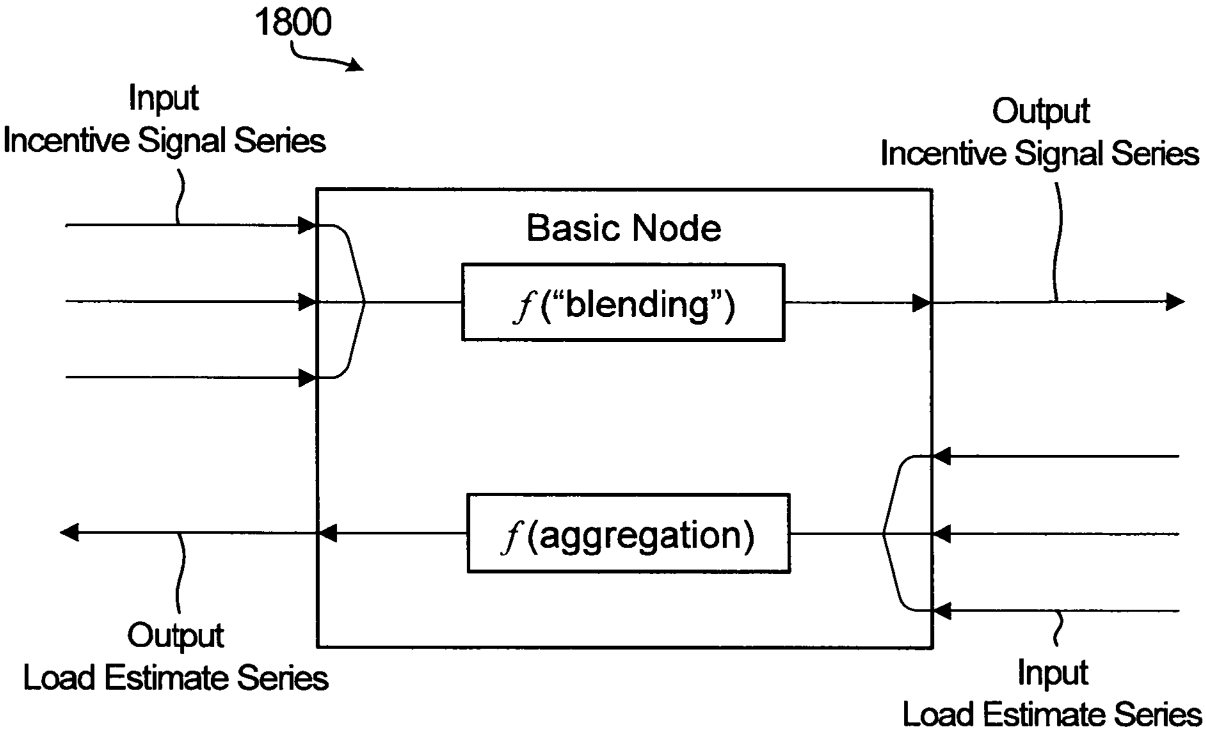

1800
Input
Incentive Signal Series
Output
Incentive Signal Series
Basic Node
f("blending")
f(aggregation)
Output
Load Estimate Series
Input
Load Estimate Series

FIG. 18

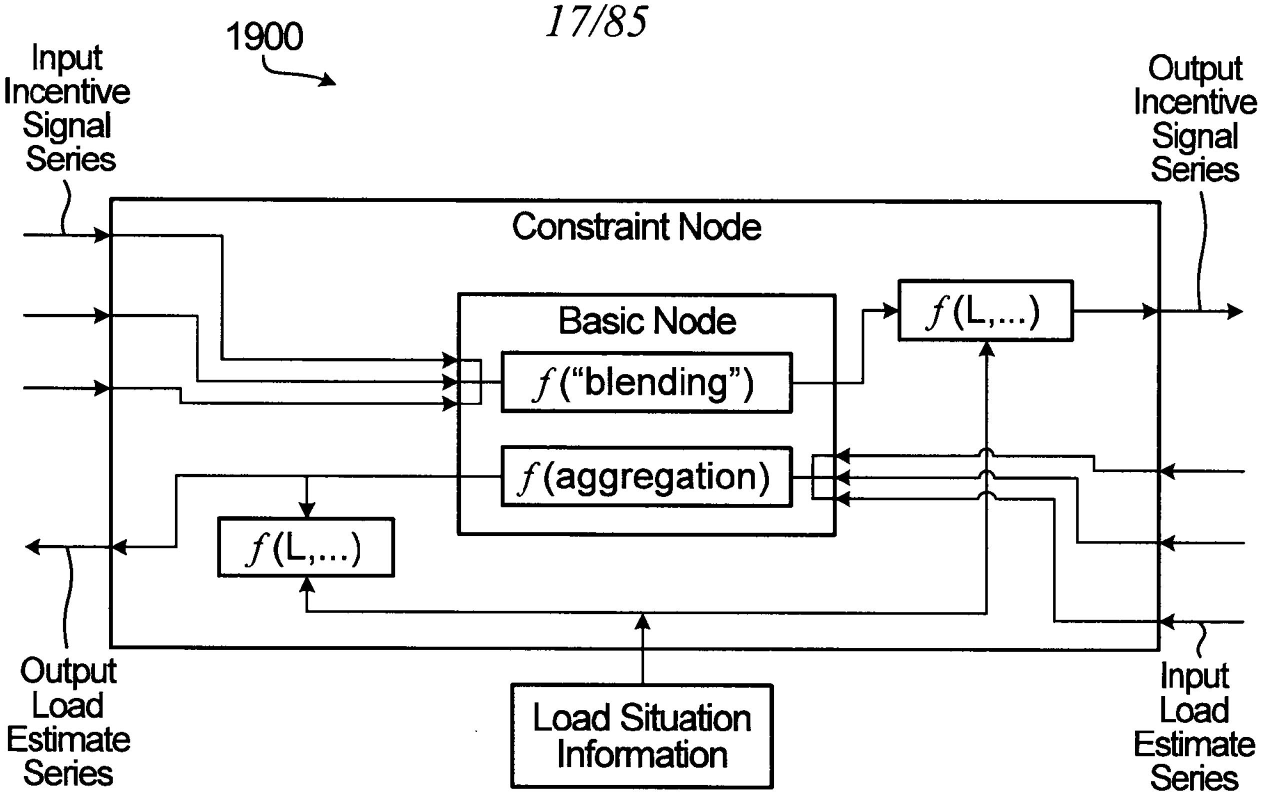

1900
Input Incentive Signal Series
Constraint Node
Basic Node
ƒ("blending")
ƒ(aggregation)
ƒ(L,...)
ƒ(L,...)
Output Incentive Signal Series
Output Load Estimate Series
Load Situation Information
Input Load Estimate Series

FIG. 19

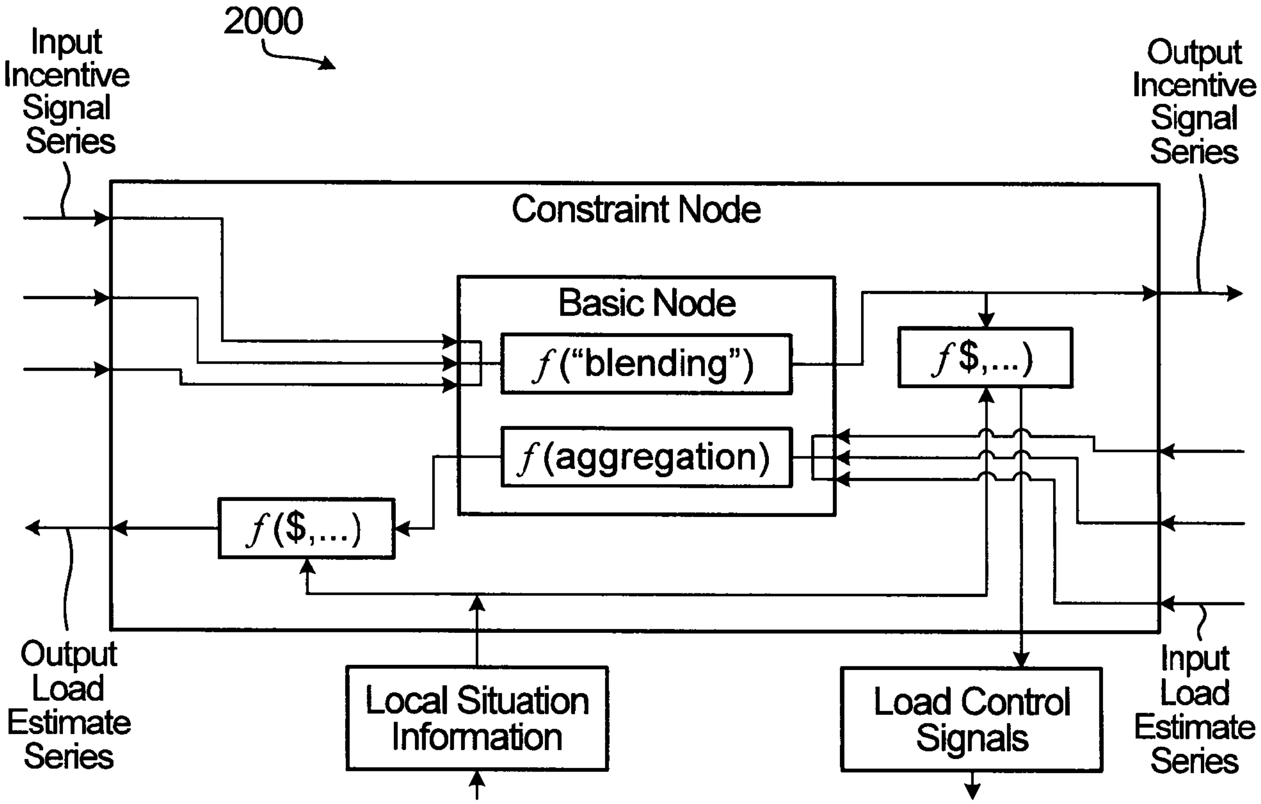

2000
Input Incentive Signal Series
Constraint Node
Basic Node
ƒ("blending")
ƒ(aggregation)
ƒ$,...)
ƒ($,...)
Output Incentive Signal Series
Output Load Estimate Series
Local Situation Information
Load Control Signals
Input Load Estimate Series

FIG. 20

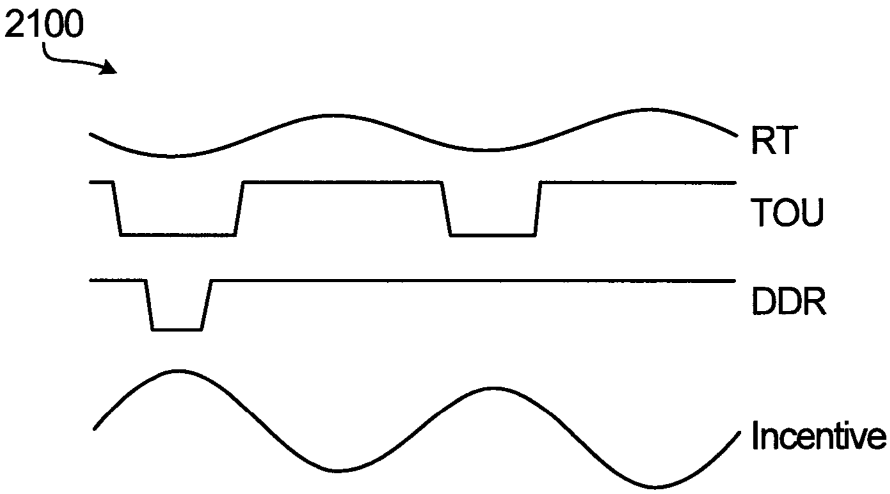

2100
RT
TOU
DDR
Incentive

FIG. 21

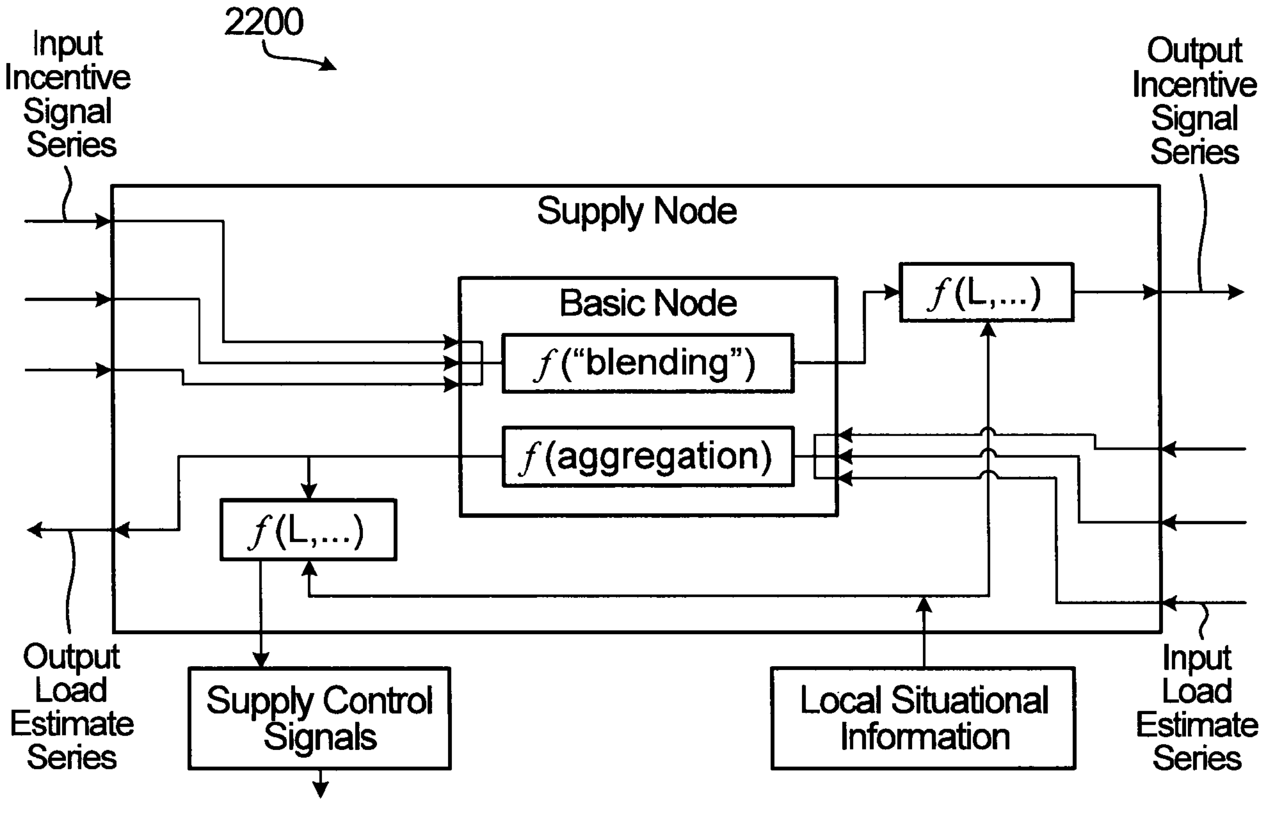

2200
Input Incentive Signal Series
Output Incentive Signal Series
Supply Node
Basic Node
f("blending")
f(aggregation)
f(L,...)
f(L,...)
Output Load Estimate Series
Input Load Estimate Series
Supply Control Signals
Local Situational Information

FIG. 22

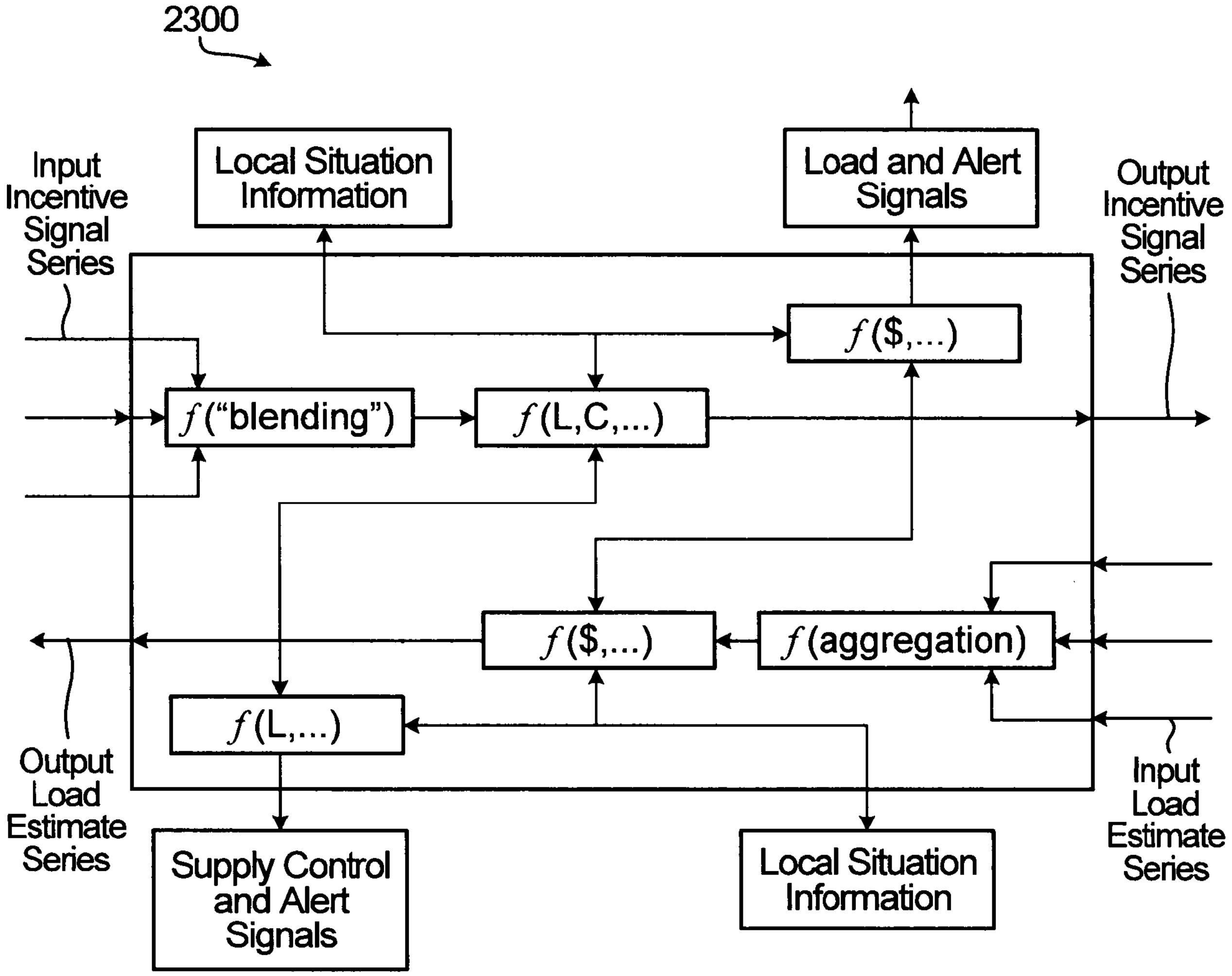

2300
Input Incentive Signal Series
Local Situation Information
Load and Alert Signals
Output Incentive Signal Series
f($,...)
f("blending")
f(L,C,...)
f($,...)
f(aggregation)
f(L,...)
Output Load Estimate Series
Input Load Estimate Series
Supply Control and Alert Signals
Local Situation Information

FIG. 23

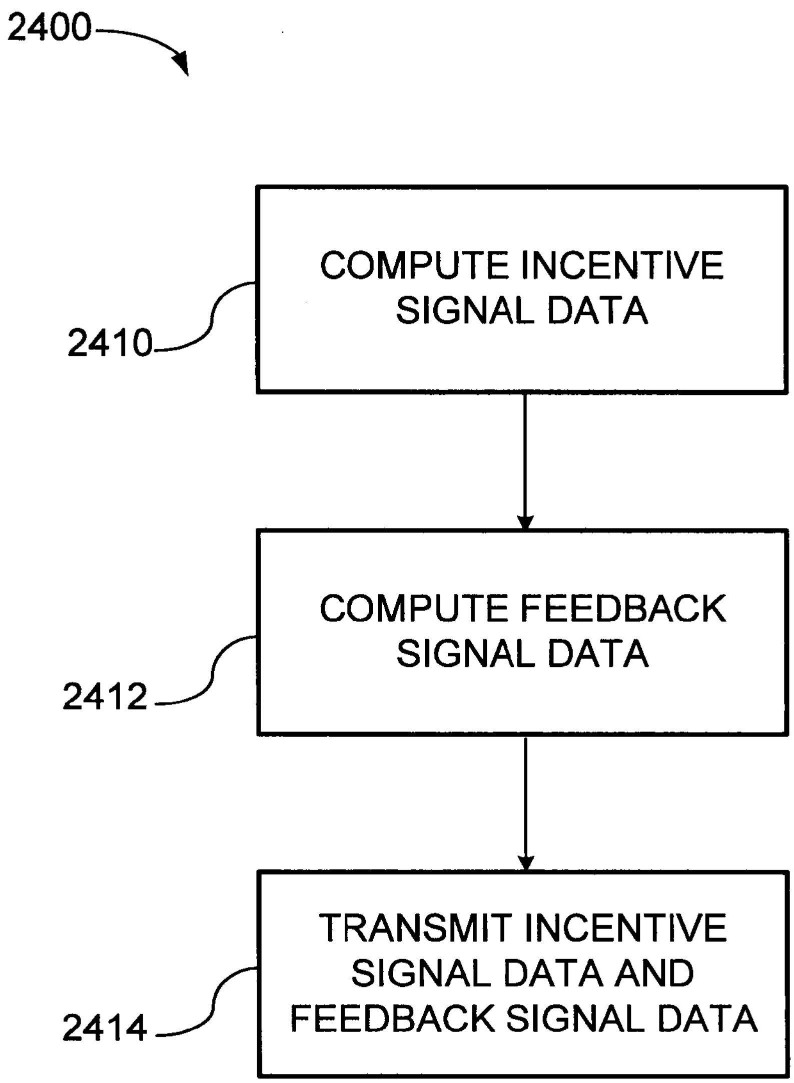

2400
COMPUTE INCENTIVE SIGNAL DATA
2410
COMPUTE FEEDBACK SIGNAL DATA
2412
TRANSMIT INCENTIVE SIGNAL DATA AND FEEDBACK SIGNAL DATA
2414

FIG. 24

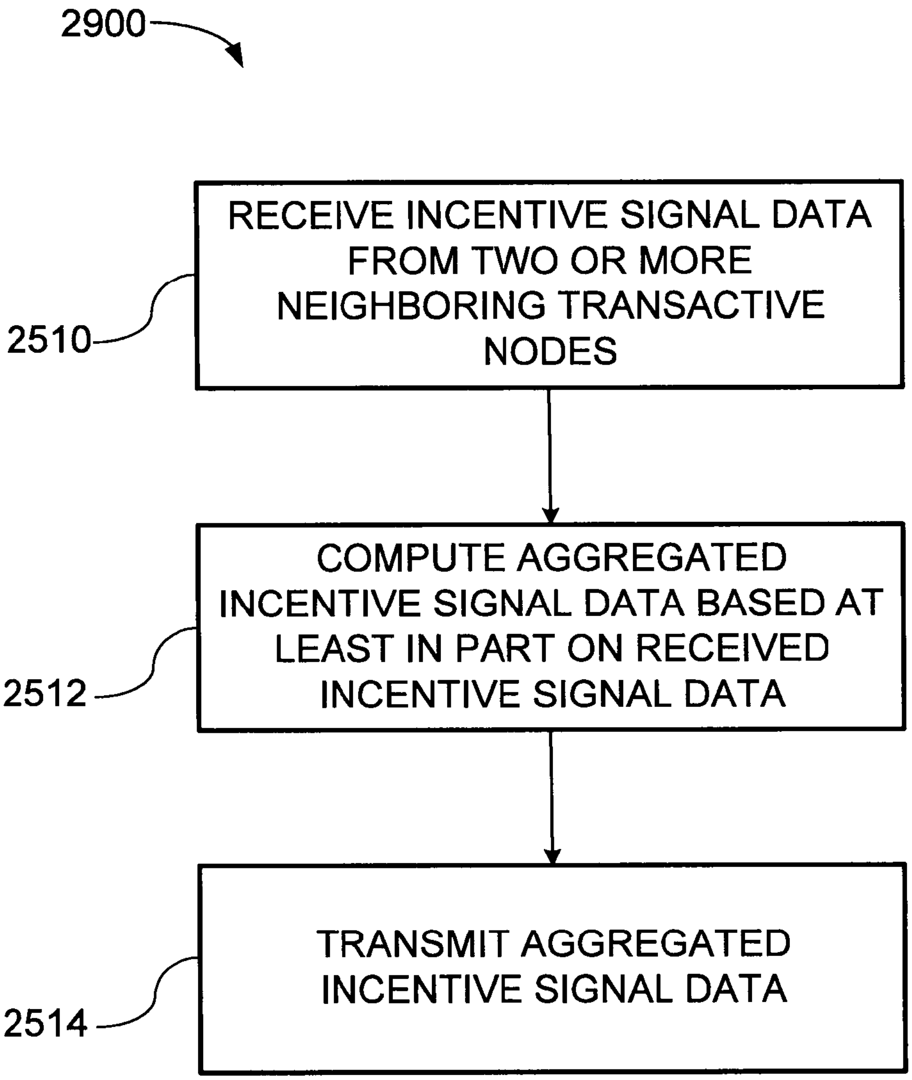

2900
RECEIVE INCENTIVE SIGNAL DATA FROM TWO OR MORE NEIGHBORING TRANSACTIVE NODES
2510
COMPUTE AGGREGATED INCENTIVE SIGNAL DATA BASED AT LEAST IN PART ON RECEIVED INCENTIVE SIGNAL DATA
2512
TRANSMIT AGGREGATED INCENTIVE SIGNAL DATA
2514

FIG. 25

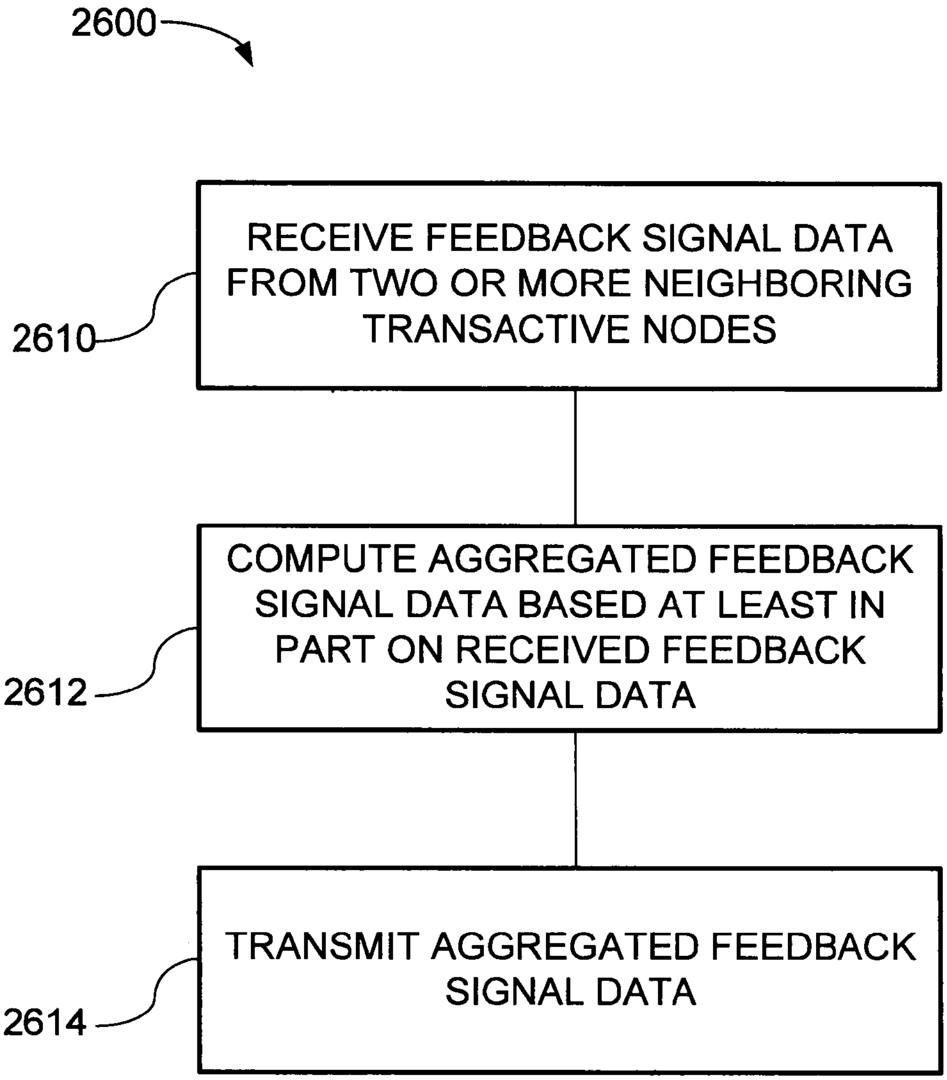

2600
RECEIVE FEEDBACK SIGNAL DATA FROM TWO OR MORE NEIGHBORING TRANSACTIVE NODES
2610
COMPUTE AGGREGATED FEEDBACK SIGNAL DATA BASED AT LEAST IN PART ON RECEIVED FEEDBACK SIGNAL DATA
2612
TRANSMIT AGGREGATED FEEDBACK SIGNAL DATA
2614

FIG. 26

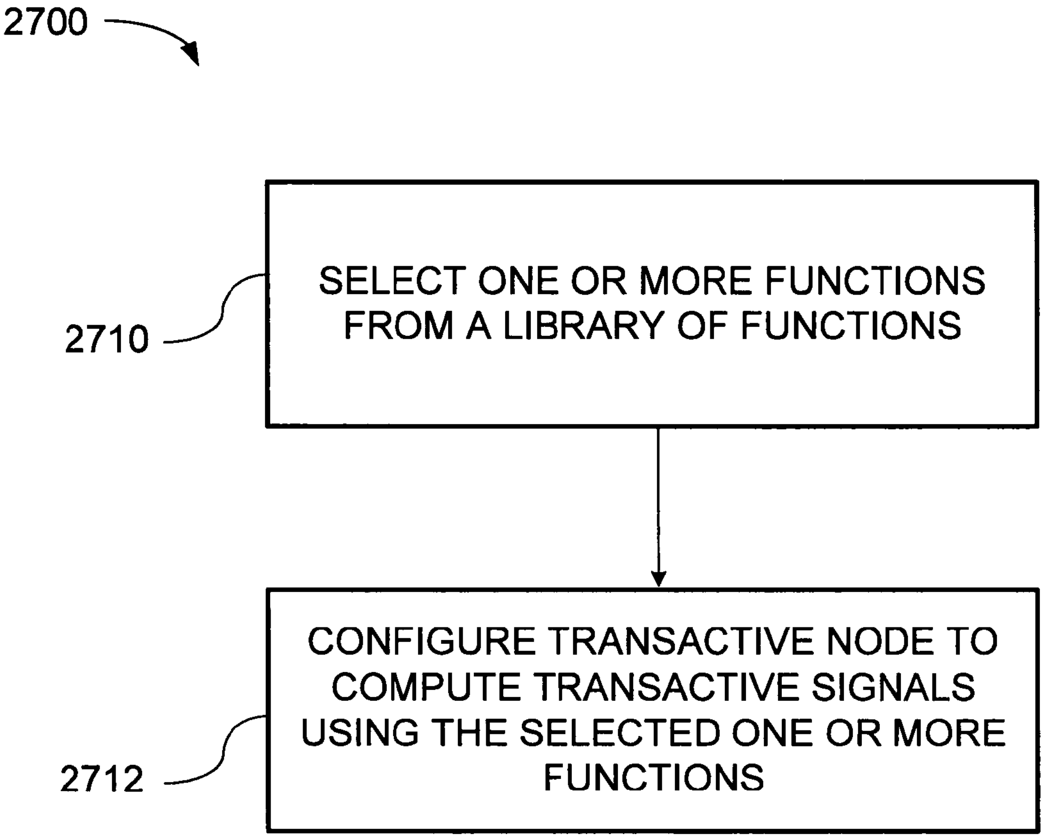

2700
2710
SELECT ONE OR MORE FUNCTIONS FROM A LIBRARY OF FUNCTIONS
2712
CONFIGURE TRANSACTIVE NODE TO COMPUTE TRANSACTIVE SIGNALS USING THE SELECTED ONE OR MORE FUNCTIONS

FIG. 27

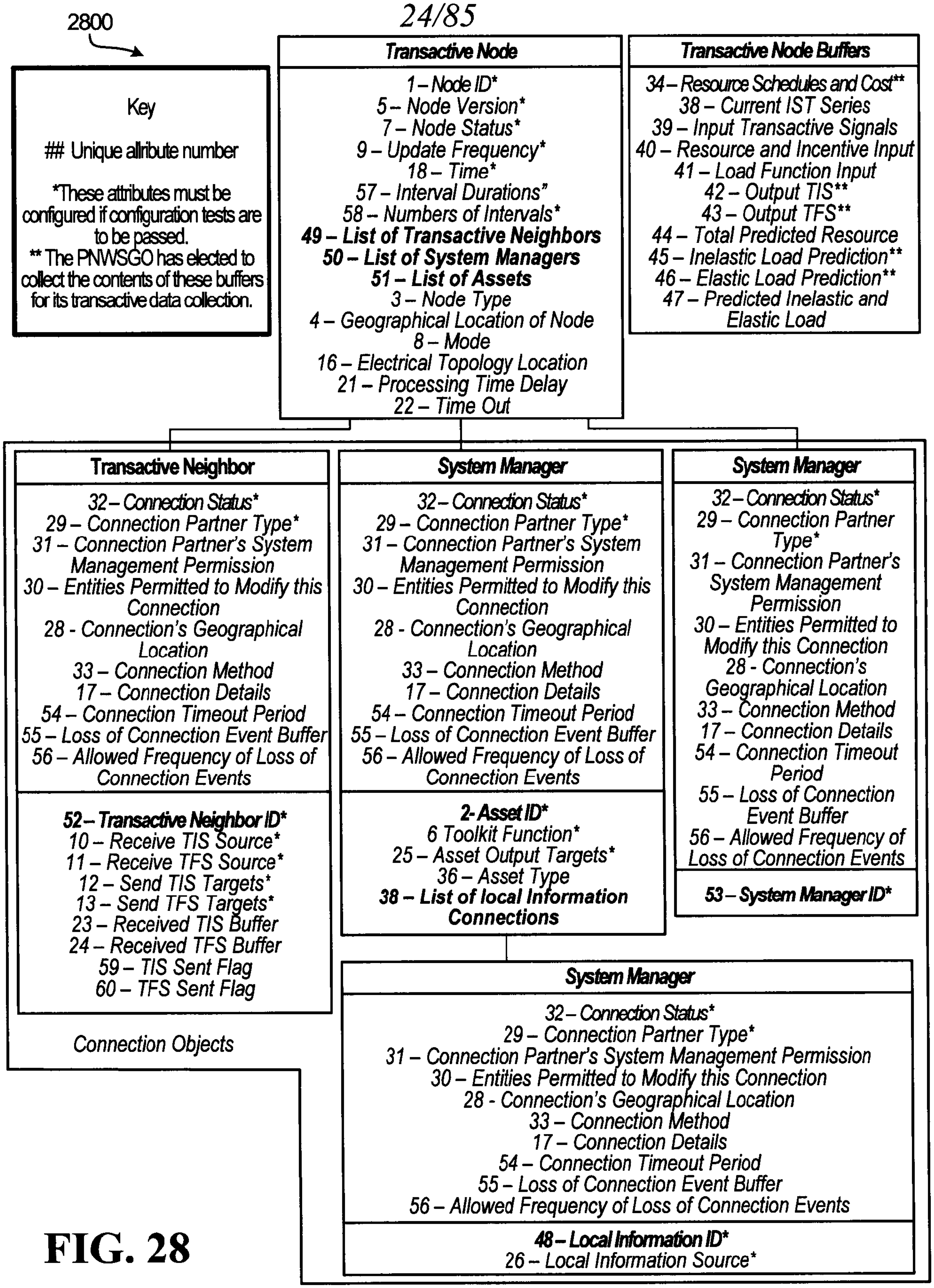
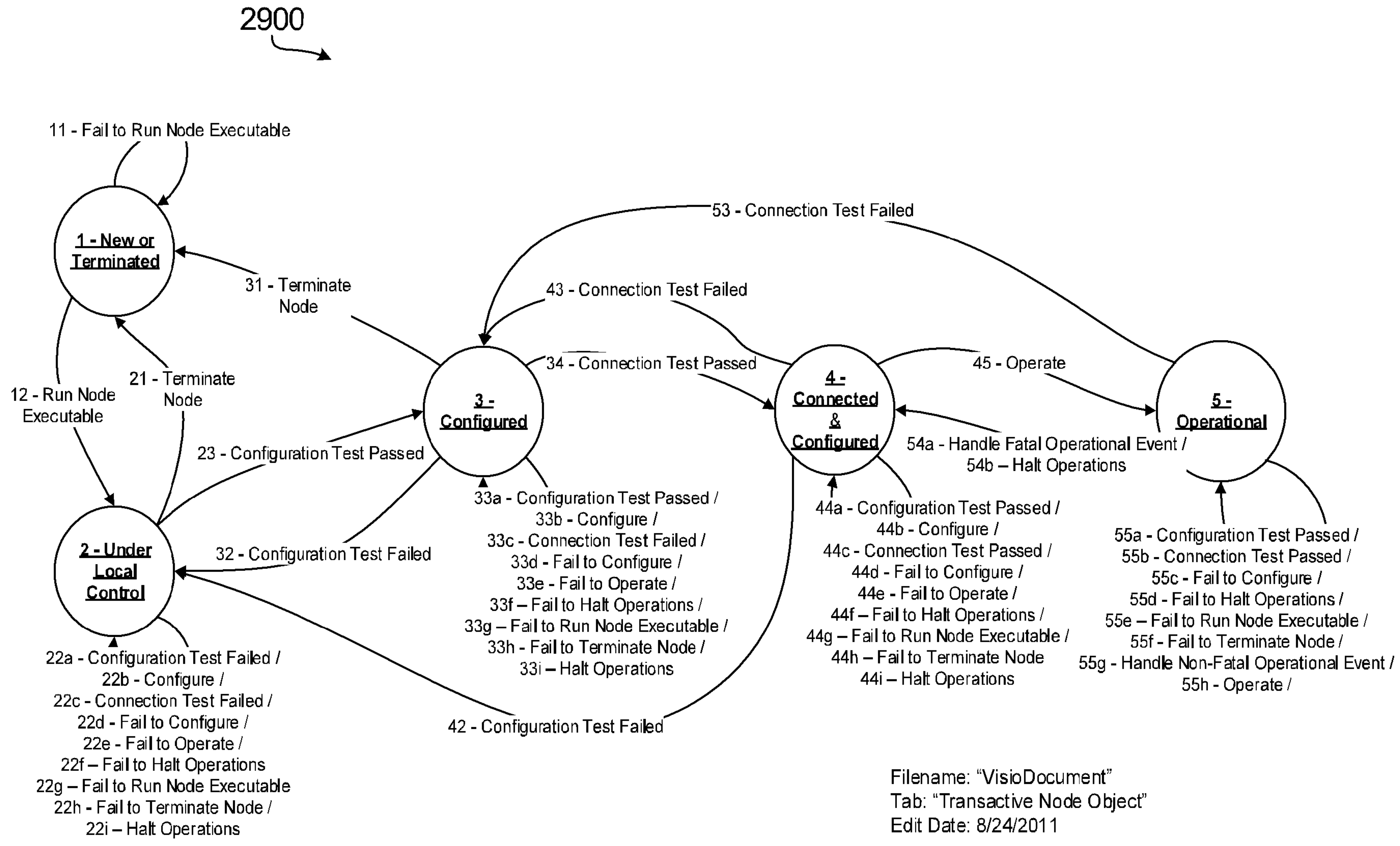

2800
Key
Unique allribute number
*These attributes must be configured if configuration tests are to be passed.
** The PNWSGO has elected to collect the contents of these buffers for its transactive data collection.
Transactive Node
1 – Node ID*
5 – Node Version*
7 – Node Status*
9 – Update Frequency*
18 – Time*
57 – Interval Durations*
58 – Numbers of Intervals*
49 – List of Transactive Neighbors
50 – List of System Managers
51 – List of Assets
3 – Node Type
4 – Geographical Location of Node
8 – Mode
16 – Electrical Topology Location
21 – Processing Time Delay
22 – Time Out
Transactive Node Buffers
34 – Resource Schedules and Cost**
38 – Current IST Series
39 – Input Transactive Signals
40 – Resource and Incentive Input
41 – Load Function Input
42 – Output TIS**
43 – Output TFS**
44 – Total Predicted Resource
45 – Inelastic Load Prediction**
46 – Elastic Load Prediction**
47 – Predicted Inelastic and Elastic Load
Transactive Neighbor
32 – Connection Status*
29 – Connection Partner Type*
31 – Connection Partner's System Management Permission
30 – Entities Permitted to Modify this Connection
28 - Connection's Geographical Location
33 – Connection Method
17 – Connection Details
54 – Connection Timeout Period
55 – Loss of Connection Event Buffer
56 – Allowed Frequency of Loss of Connection Events
52 – Transactive Neighbor ID*
10 – Receive TIS Source*
11 – Receive TFS Source*
12 – Send TIS Targets*
13 – Send TFS Targets*
23 – Received TIS Buffer
24 – Received TFS Buffer
59 – TIS Sent Flag
60 – TFS Sent Flag
System Manager
32 – Connection Status*
29 – Connection Partner Type*
31 – Connection Partner's System Management Permission
30 – Entities Permitted to Modify this Connection
28 - Connection's Geographical Location
33 – Connection Method
17 – Connection Details
54 – Connection Timeout Period
55 – Loss of Connection Event Buffer
56 – Allowed Frequency of Loss of Connection Events
2- Asset ID*
6 Toolkit Function*
25 – Asset Output Targets*
36 – Asset Type
38 – List of local Information Connections
System Manager
32 – Connection Status*
29 – Connection Partner Type*
31 – Connection Partner's System Management Permission
30 – Entities Permitted to Modify this Connection
28 - Connection's Geographical Location
33 – Connection Method
17 – Connection Details
54 – Connection Timeout Period
55 – Loss of Connection Event Buffer
56 – Allowed Frequency of Loss of Connection Events
53 – System Manager ID*
System Manager
32 – Connection Status*
29 – Connection Partner Type*
31 – Connection Partner's System Management Permission
30 – Entities Permitted to Modify this Connection
28 - Connection's Geographical Location
33 – Connection Method
17 – Connection Details
54 – Connection Timeout Period
55 – Loss of Connection Event Buffer
56 – Allowed Frequency of Loss of Connection Events
48 – Local Information ID*
26 – Local Information Source*
Connection Objects

FIG. 28



Filename: “VisioDocument”
Tab: “Transactive Node Object”
Edit Date: 8/24/2011

FIG. 29

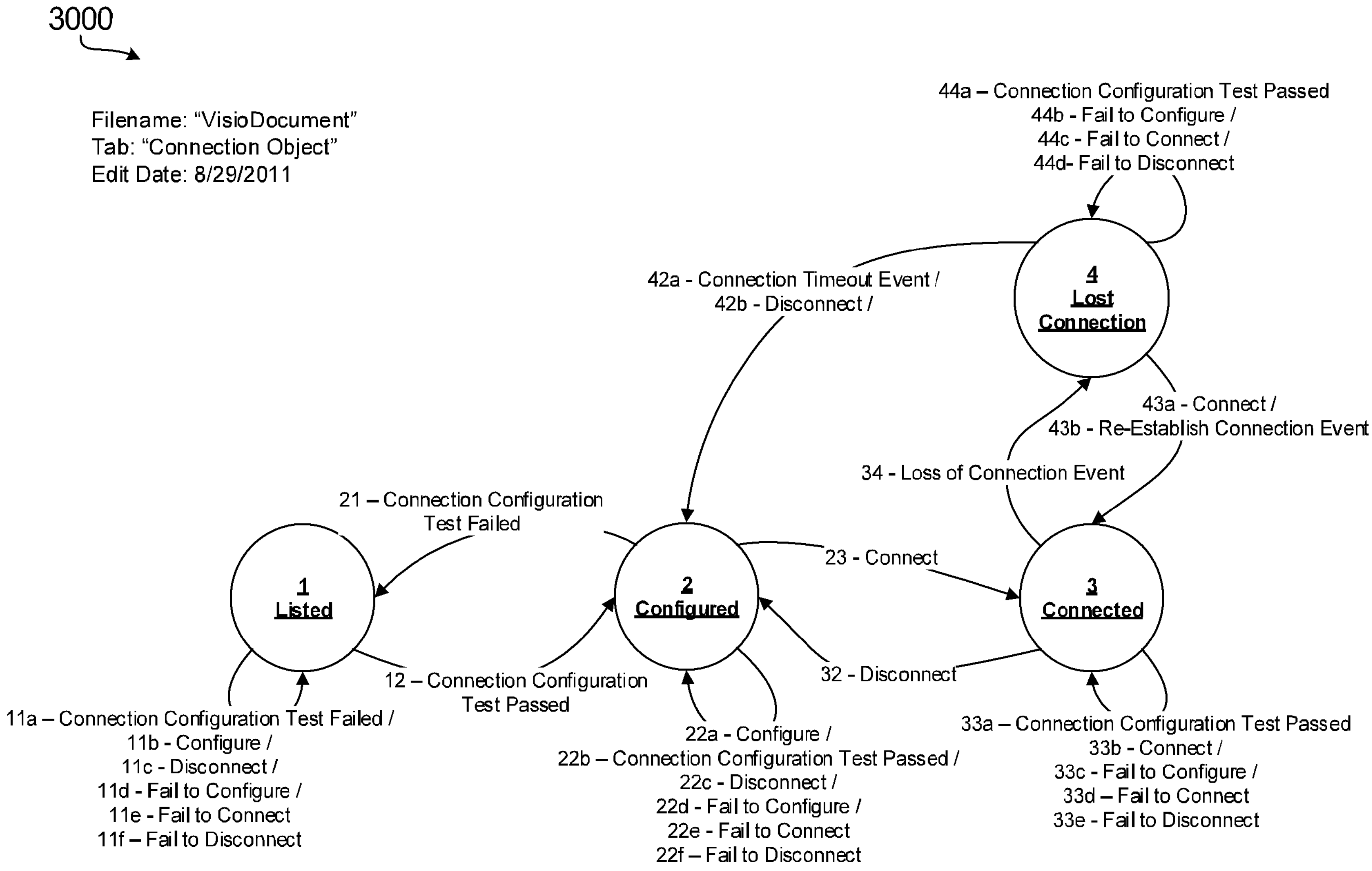

3000
Filename: "VisioDocument"
Tab: "Connection Object"
Edit Date: 8/29/2011
1
Listed
2
Configured
3
Connected
4
Lost
Connection
11a – Connection Configuration Test Failed /
11b - Configure /
11c - Disconnect /
11d - Fail to Configure /
11e - Fail to Connect
11f – Fail to Disconnect
12 – Connection Configuration
Test Passed
21 – Connection Configuration
Test Failed
22a - Configure /
22b – Connection Configuration Test Passed /
22c - Disconnect /
22d - Fail to Configure /
22e - Fail to Connect
22f – Fail to Disconnect
23 - Connect
32 - Disconnect
33a – Connection Configuration Test Passed
33b - Connect /
33c - Fail to Configure /
33d – Fail to Connect
33e - Fail to Disconnect
34 - Loss of Connection Event
42a - Connection Timeout Event /
42b - Disconnect /
43a - Connect /
43b - Re-Establish Connection Event
44a – Connection Configuration Test Passed
44b - Fail to Configure /
44c - Fail to Connect /
44d- Fail to Disconnect

FIG. 30

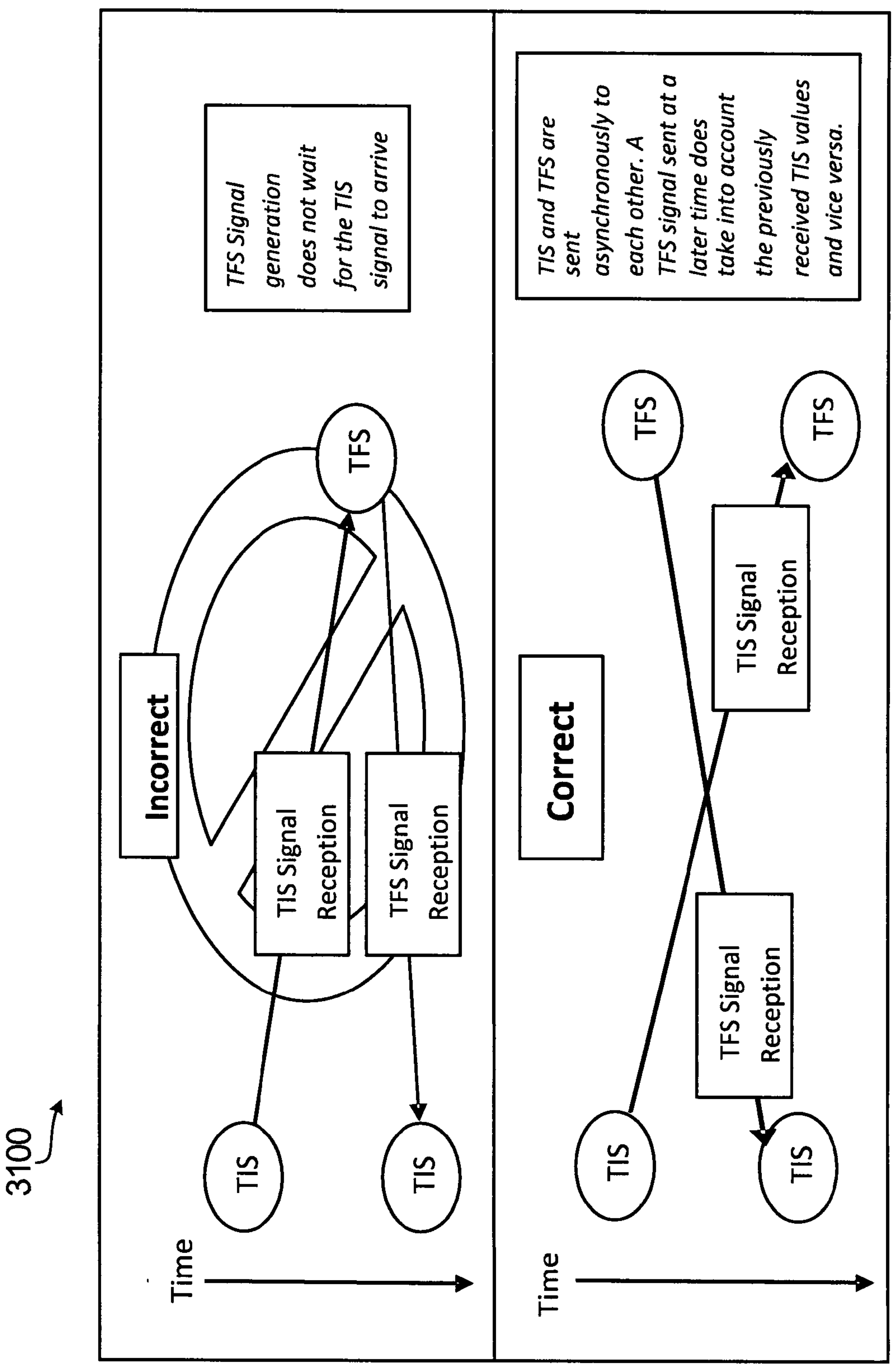

3100
Incorrect
Time
TIS
TIS
TIS Signal Reception
TFS Signal Reception
TFS
TFS Signal generation does not wait for the TIS signal to arrive
Correct
Time
TIS
TIS
TFS Signal Reception
TIS Signal Reception
TFS
TFS
TIS and TFS are sent asynchronously to each other. A TFS signal sent at a later time does take into account the previously received TIS values and vice versa.

FIG. 31

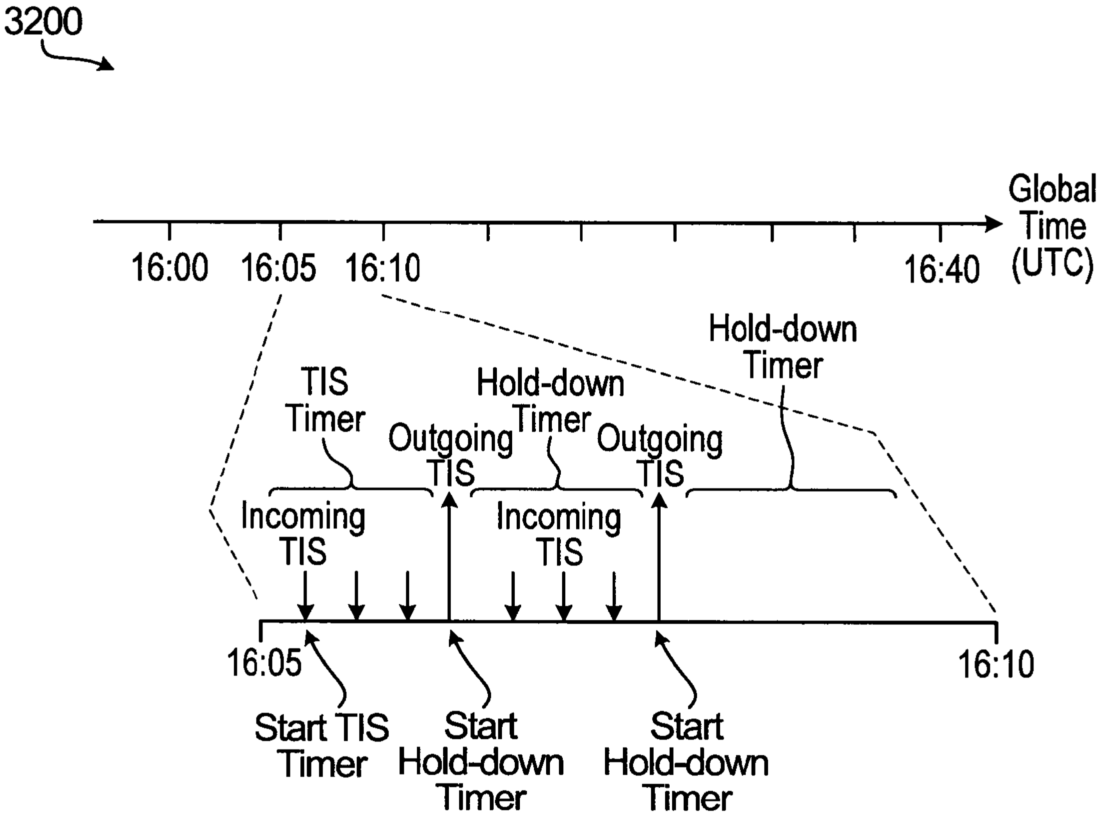

3200
Global Time (UTC)
16:00
16:05
16:10
16:40
TIS Timer
Hold-down Timer
Hold-down Timer
Outgoing TIS
Outgoing TIS
Incoming TIS
Incoming TIS
16:05
16:10
Start TIS Timer
Start Hold-down Timer
Start Hold-down Timer

FIG. 32

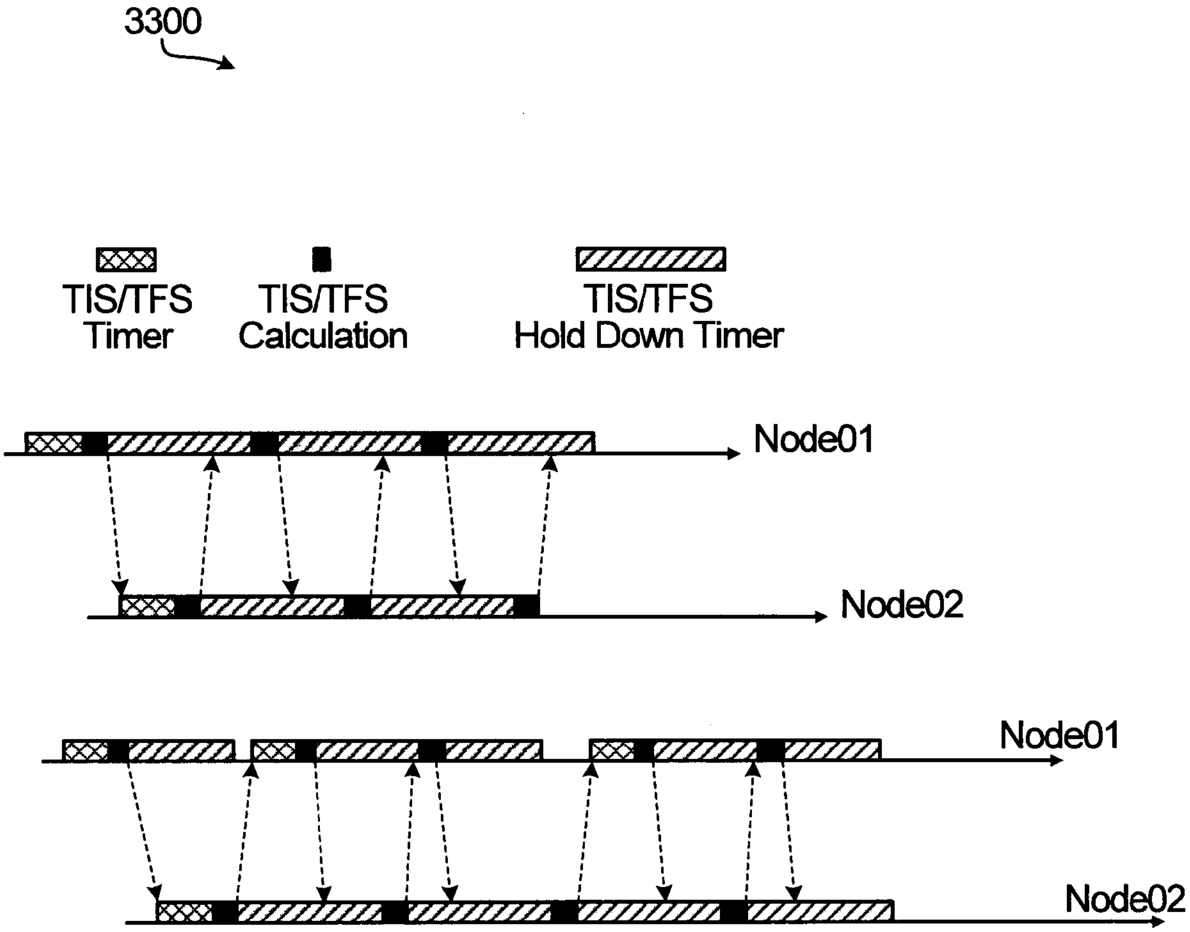

3300
TIS/TFS Timer
TIS/TFS Calculation
TIS/TFS Hold Down Timer
Node01
Node02
Node01
Node02

FIG. 33

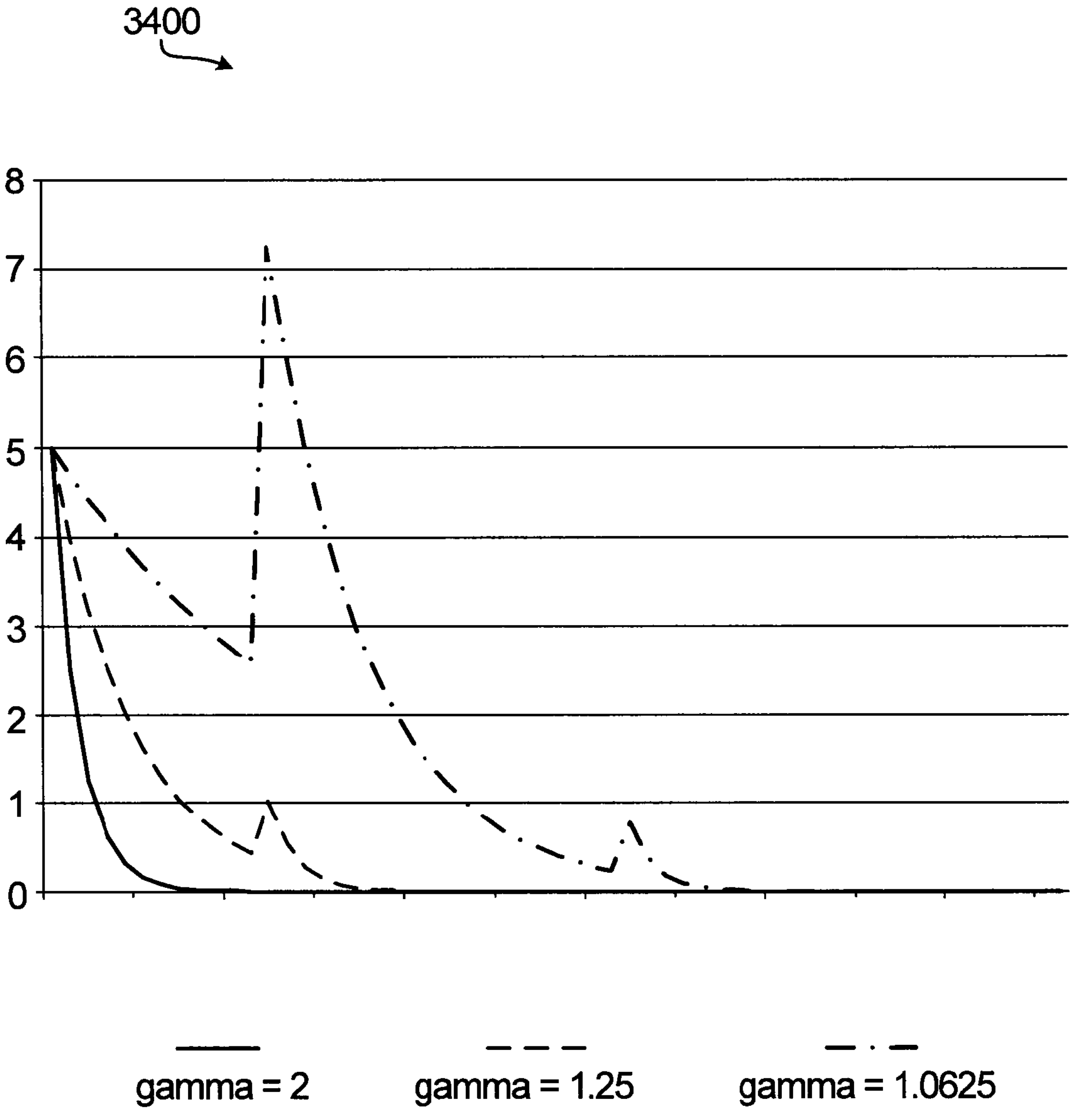


FIG. 34

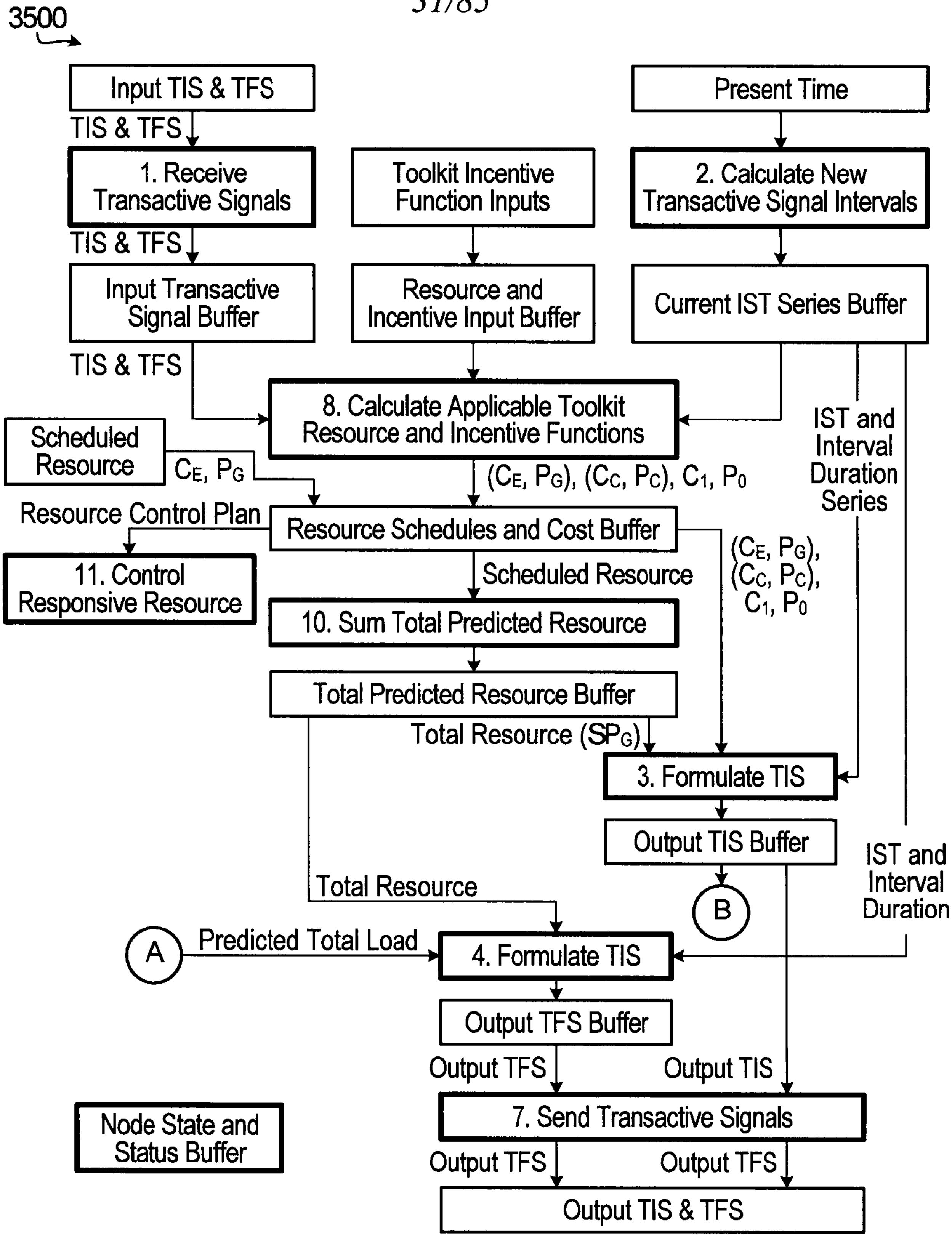

3500
Input TIS & TFS
TIS & TFS
1. Receive Transactive Signals
TIS & TFS
Input Transactive Signal Buffer
TIS & TFS
Toolkit Incentive Function Inputs
Resource and Incentive Input Buffer
Present Time
2. Calculate New Transactive Signal Intervals
Current IST Series Buffer
8. Calculate Applicable Toolkit Resource and Incentive Functions
IST and Interval Duration Series
Scheduled Resource
C_E, P_G
(C_E, P_G), (C_C, P_C), C_1, P_0
Resource Control Plan
Resource Schedules and Cost Buffer
11. Control Responsive Resource
Scheduled Resource
(C_E, P_G), (C_C, P_C), C_1, P_0
10. Sum Total Predicted Resource
Total Predicted Resource Buffer
Total Resource (SP_G)
3. Formulate TIS
Output TIS Buffer
IST and Interval Duration
Total Resource
B
A
Predicted Total Load
4. Formulate TIS
Output TFS Buffer
Output TFS
Output TIS
Node State and Status Buffer
7. Send Transactive Signals
Output TFS
Output TFS
Output TIS & TFS

FIG. 35A

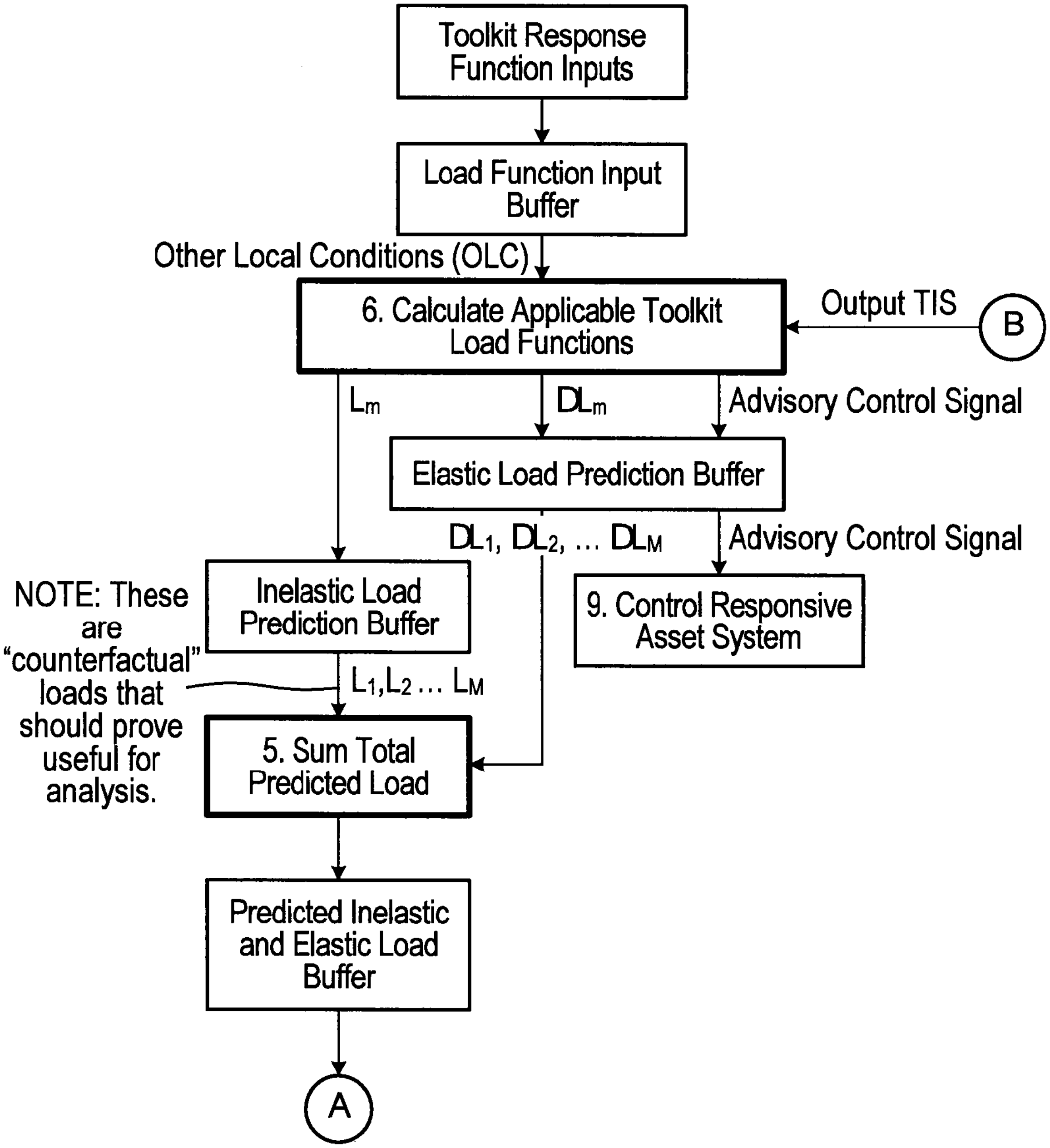

Toolkit Response Function Inputs
Load Function Input Buffer
Other Local Conditions (OLC)
6. Calculate Applicable Toolkit Load Functions
Output TIS
B
Lm
DLm
Advisory Control Signal
Elastic Load Prediction Buffer
DL1, DL2, ... DLM
Advisory Control Signal
Inelastic Load Prediction Buffer
9. Control Responsive Asset System
NOTE: These are "counterfactual" loads that should prove useful for analysis.
L1,L2 ... LM
5. Sum Total Predicted Load
Predicted Inelastic and Elastic Load Buffer
A

FIG. 35B

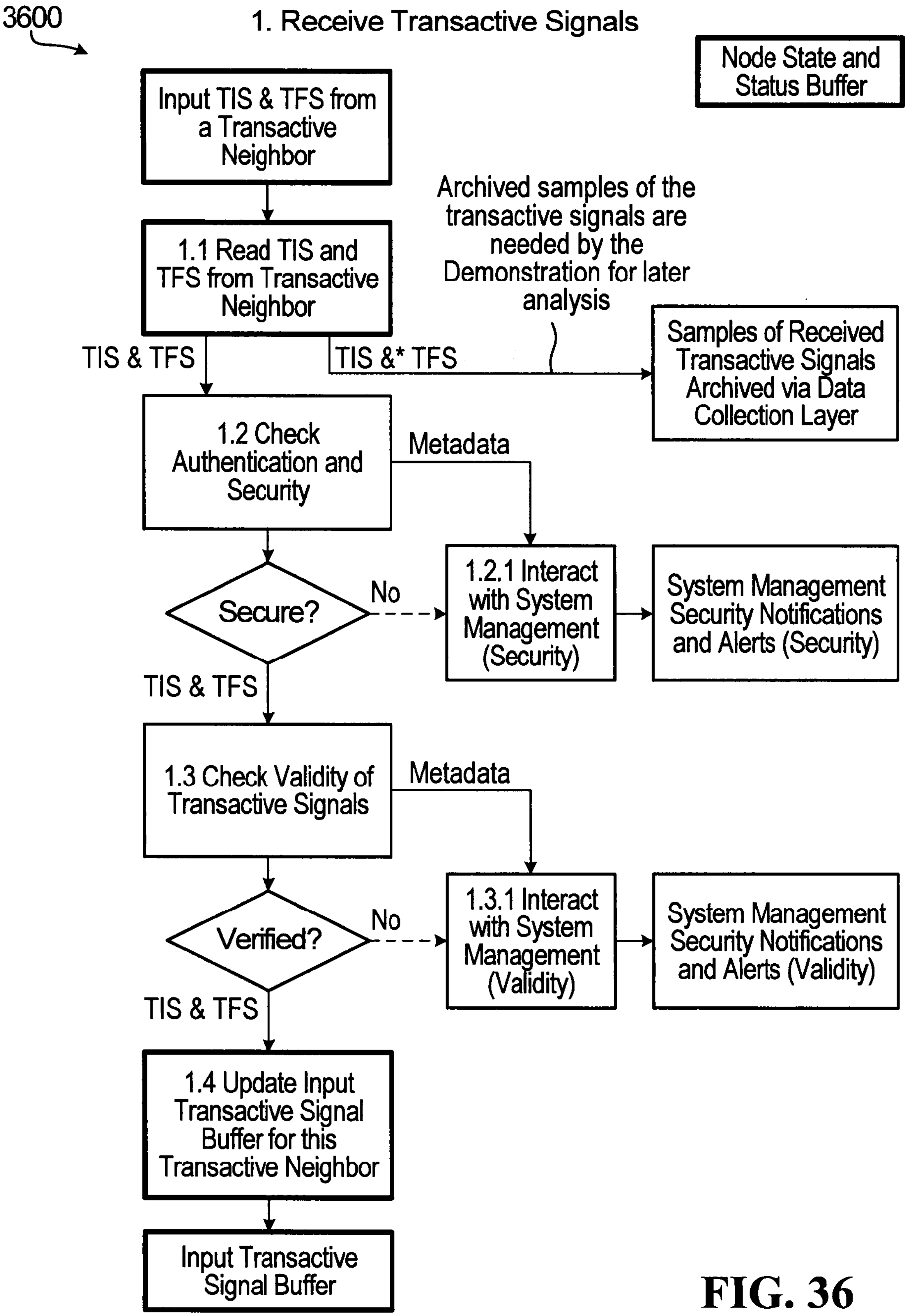

3600
1. Receive Transactive Signals
Node State and Status Buffer
Input TIS & TFS from a Transactive Neighbor
Archived samples of the transactive signals are needed by the Demonstration for later analysis
1.1 Read TIS and TFS from Transactive Neighbor
TIS & TFS
TIS &* TFS
Samples of Received Transactive Signals Archived via Data Collection Layer
1.2 Check Authentication and Security
Metadata
Secure?
No
1.2.1 Interact with System Management (Security)
System Management Security Notifications and Alerts (Security)
TIS & TFS
1.3 Check Validity of Transactive Signals
Metadata
Verified?
No
1.3.1 Interact with System Management (Validity)
System Management Security Notifications and Alerts (Validity)
TIS & TFS
1.4 Update Input Transactive Signal Buffer for this Transactive Neighbor
Input Transactive Signal Buffer

FIG. 36

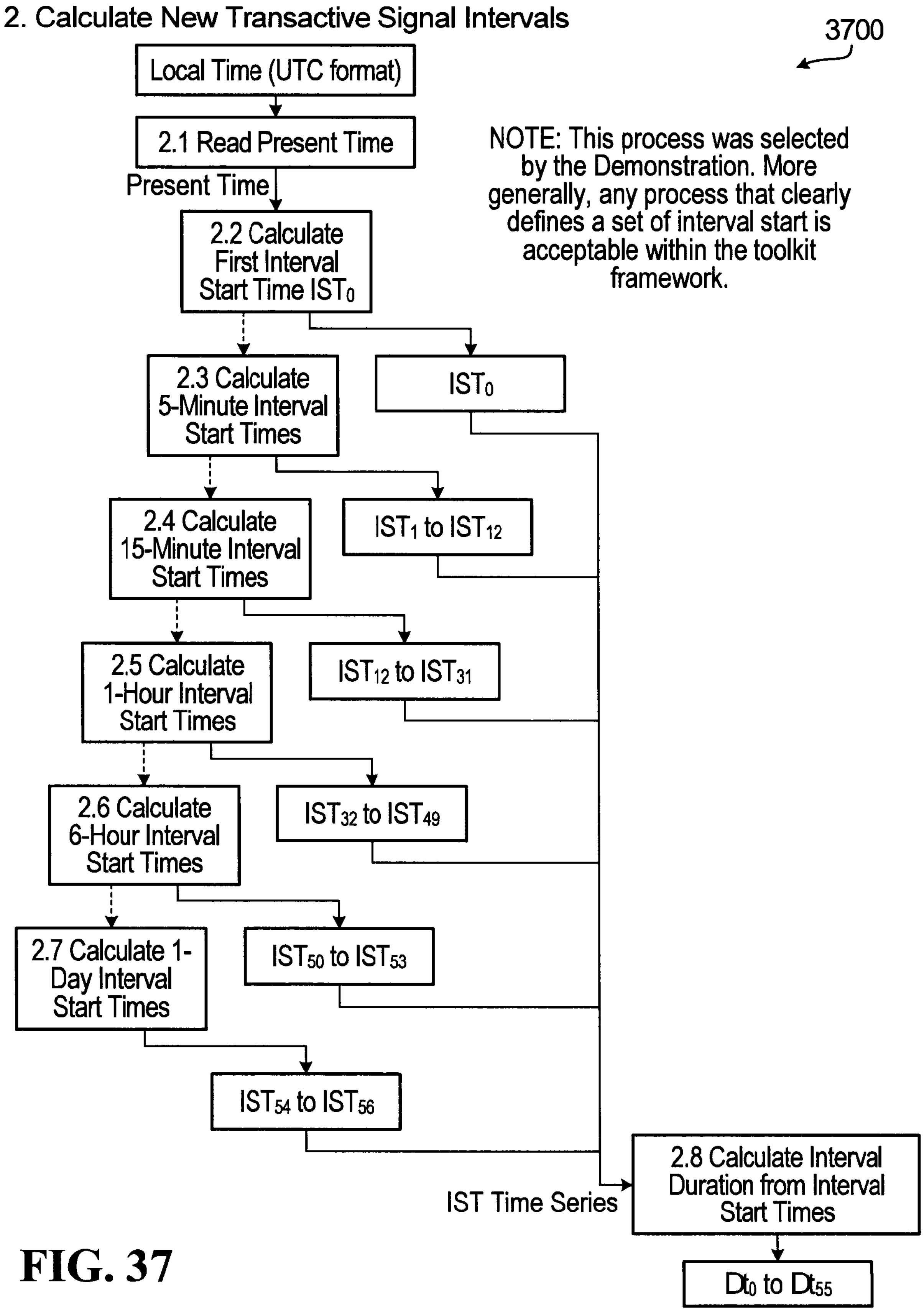

2. Calculate New Transactive Signal Intervals
3700
Local Time (UTC format)
2.1 Read Present Time
Present Time
NOTE: This process was selected by the Demonstration. More generally, any process that clearly defines a set of interval start is acceptable within the toolkit framework.
2.2 Calculate First Interval Start Time IST$_0$
2.3 Calculate 5-Minute Interval Start Times
IST$_0$
2.4 Calculate 15-Minute Interval Start Times
IST$_1$ to IST$_{12}$
2.5 Calculate 1-Hour Interval Start Times
IST$_{12}$ to IST$_{31}$
2.6 Calculate 6-Hour Interval Start Times
IST$_{32}$ to IST$_{49}$
2.7 Calculate 1-Day Interval Start Times
IST$_{50}$ to IST$_{53}$
IST$_{54}$ to IST$_{56}$
IST Time Series
2.8 Calculate Interval Duration from Interval Start Times
Dt$_0$ to Dt$_{55}$

FIG. 37

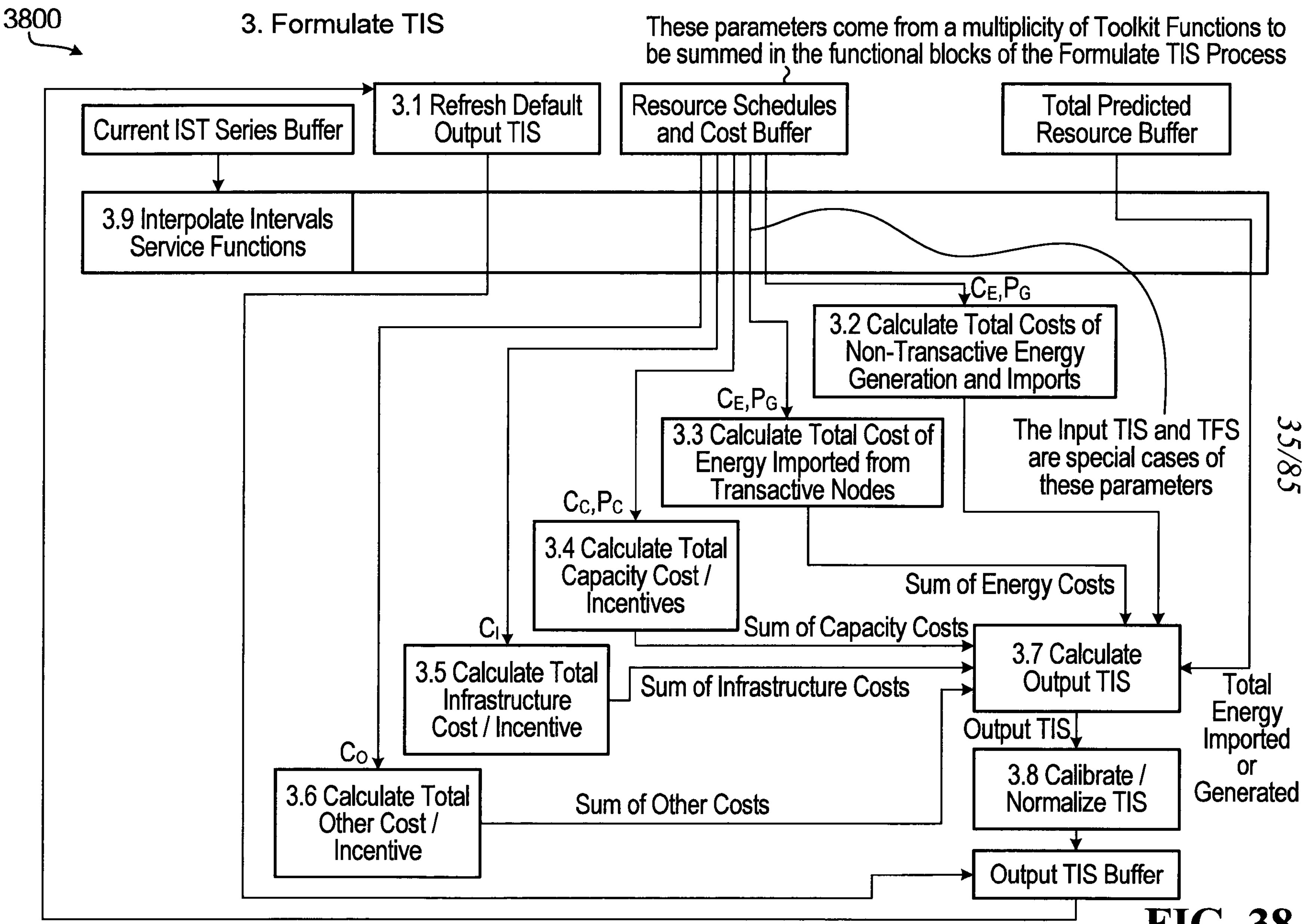

3800
3. Formulate TIS
These parameters come from a multiplicity of Toolkit Functions to be summed in the functional blocks of the Formulate TIS Process
Current IST Series Buffer
3.1 Refresh Default Output TIS
Resource Schedules and Cost Buffer
Total Predicted Resource Buffer
3.9 Interpolate Intervals Service Functions
C_E, P_G
3.2 Calculate Total Costs of Non-Transactive Energy Generation and Imports
C_E, P_G
3.3 Calculate Total Cost of Energy Imported from Transactive Nodes
The Input TIS and TFS are special cases of these parameters
C_C, P_C
3.4 Calculate Total Capacity Cost / Incentives
Sum of Energy Costs
Sum of Capacity Costs
C_I
3.5 Calculate Total Infrastructure Cost / Incentive
Sum of Infrastructure Costs
3.7 Calculate Output TIS
Total Energy Imported or Generated
C_O
3.6 Calculate Total Other Cost / Incentive
Sum of Other Costs
Output TIS
3.8 Calibrate / Normalize TIS
Output TIS Buffer

FIG. 38

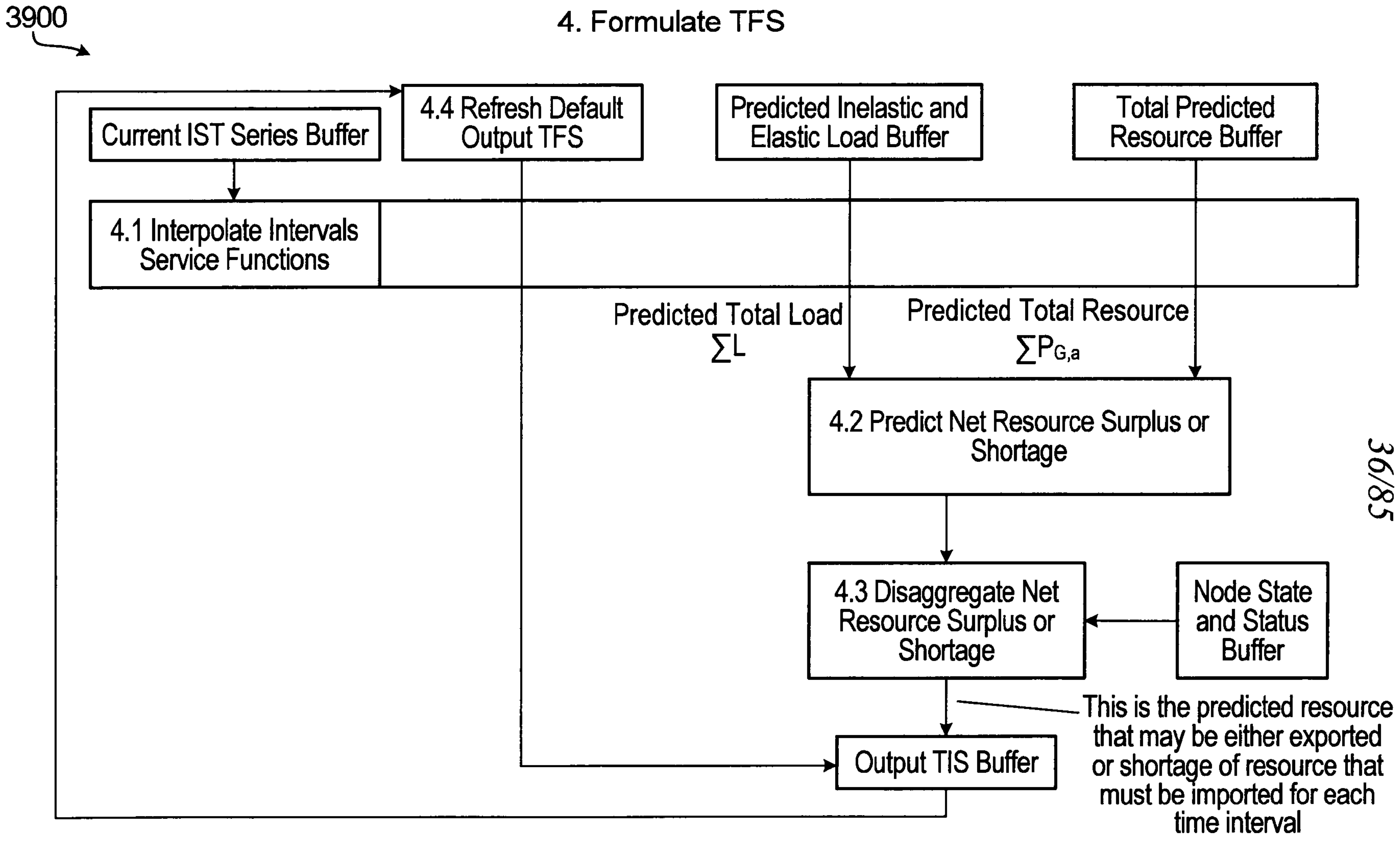
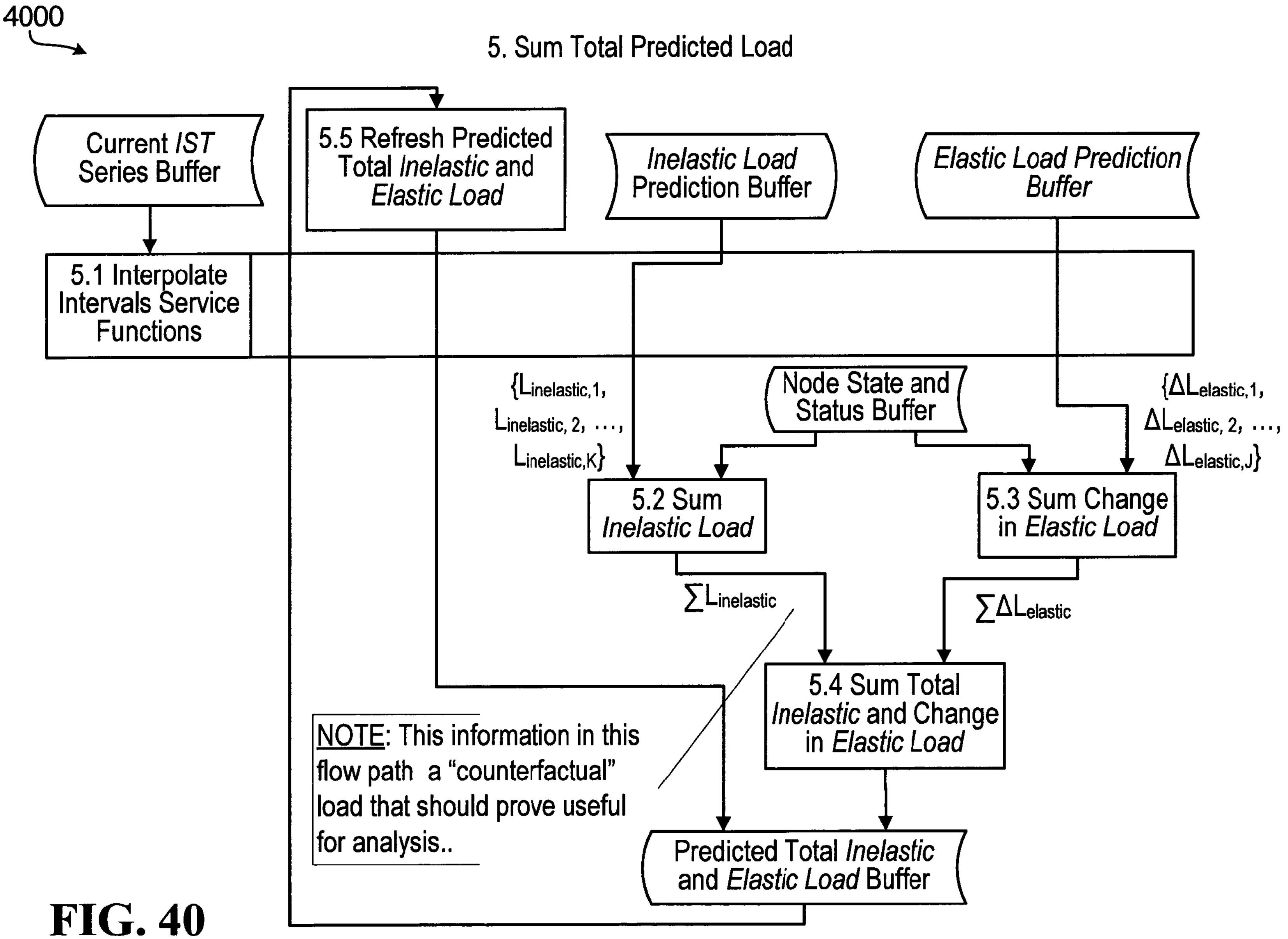

3900
4. Formulate TFS
Current IST Series Buffer
4.4 Refresh Default Output TFS
Predicted Inelastic and Elastic Load Buffer
Total Predicted Resource Buffer
4.1 Interpolate Intervals Service Functions
Predicted Total Load
$\sum L$
Predicted Total Resource
$\sum P_{G,a}$
4.2 Predict Net Resource Surplus or Shortage
4.3 Disaggregate Net Resource Surplus or Shortage
Node State and Status Buffer
Output TIS Buffer
This is the predicted resource that may be either exported or shortage of resource that must be imported for each time interval

FIG. 39



NOTE: This information in this flow path a "counterfactual" load that should prove useful for analysis..

FIG. 40

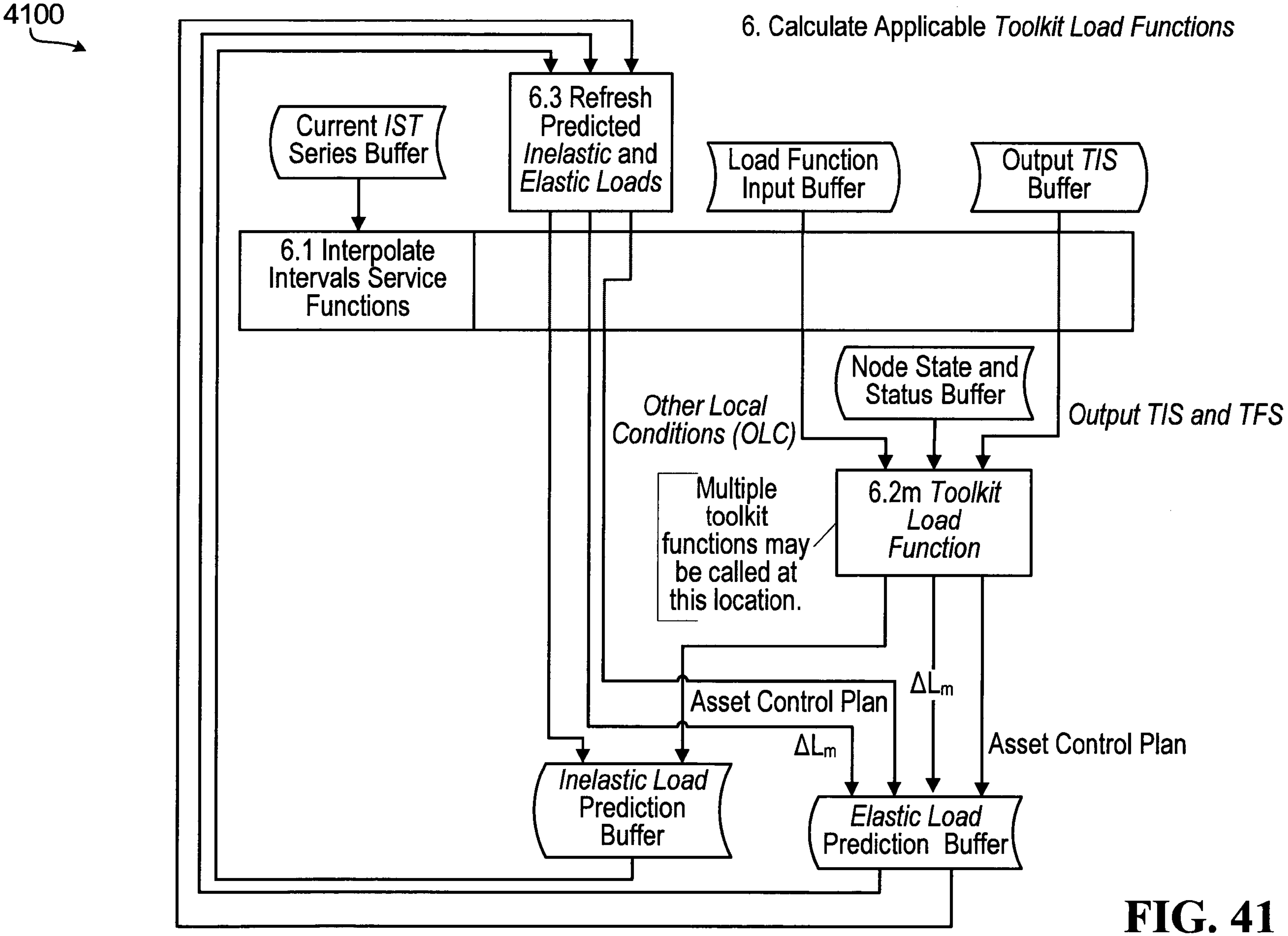

4100
6. Calculate Applicable *Toolkit Load Functions*
Current *IST* Series Buffer
6.3 Refresh Predicted *Inelastic* and *Elastic Loads*
Load Function Input Buffer
Output *TIS* Buffer
6.1 Interpolate Intervals Service Functions
Node State and Status Buffer
Other Local Conditions (OLC)
Output TIS and TFS
Multiple toolkit functions may be called at this location.
6.2m *Toolkit Load Function*
Asset Control Plan
ΔLm
ΔLm
Asset Control Plan
Inelastic Load Prediction Buffer
Elastic Load Prediction Buffer

FIG. 41

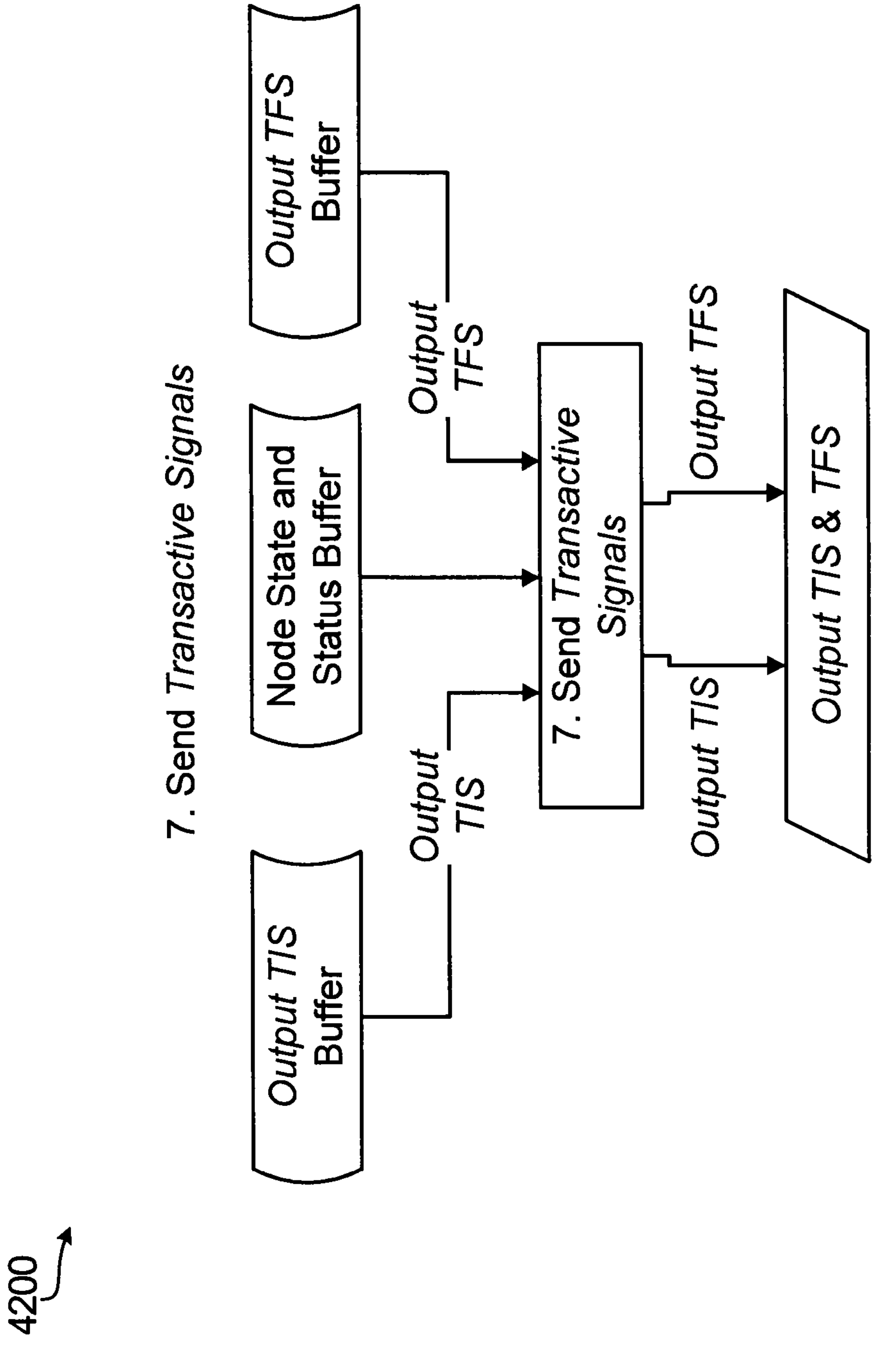
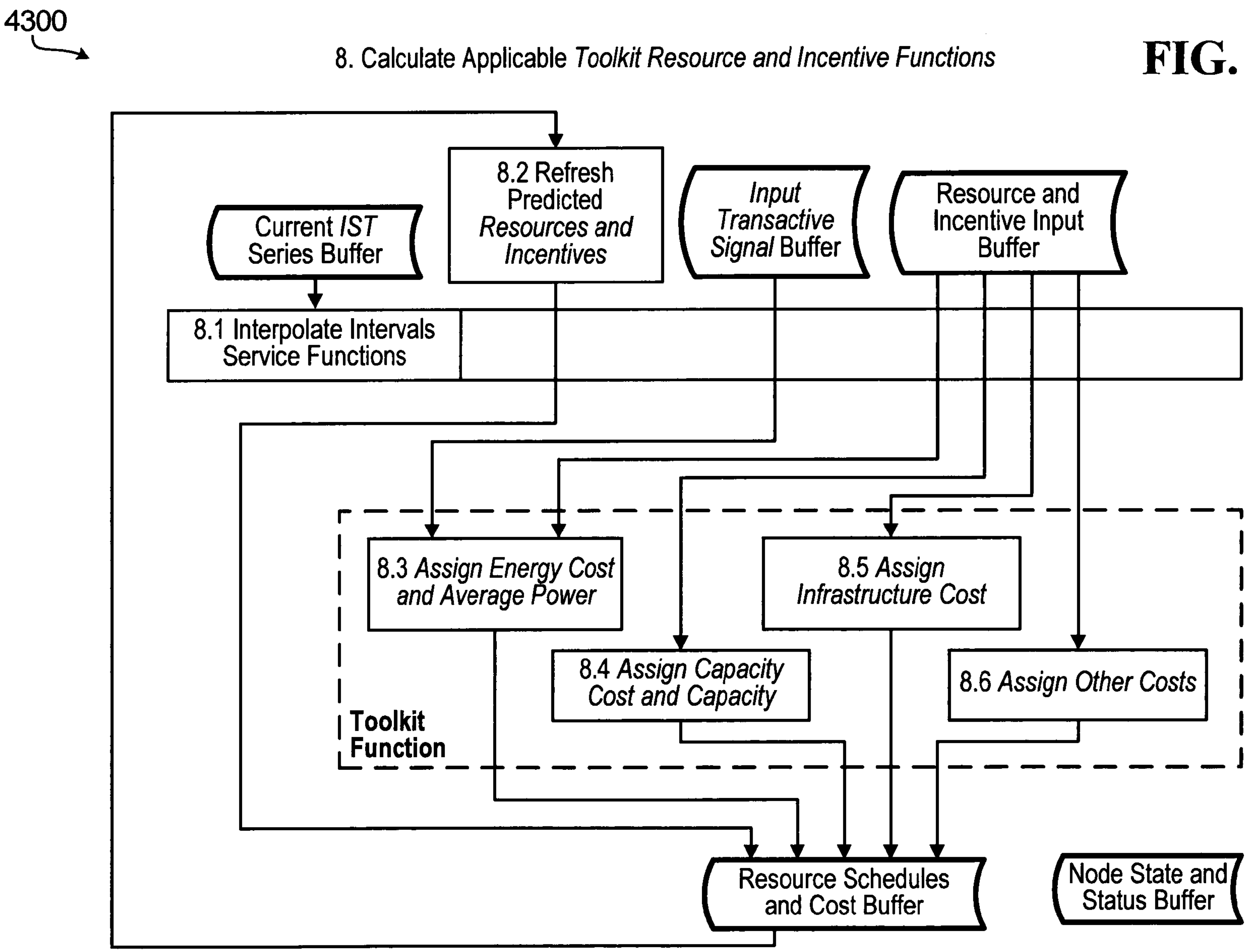

4200
7. Send *Transactive Signals*
Output TIS Buffer
Node State and Status Buffer
Output TFS Buffer
Output TIS
Output TFS
7. Send *Transactive Signals*
Output TIS
Output TFS
Output TIS & *TFS*

FIG. 42

FIG. 43

8. Calculate Applicable *Toolkit Resource and Incentive Functions*



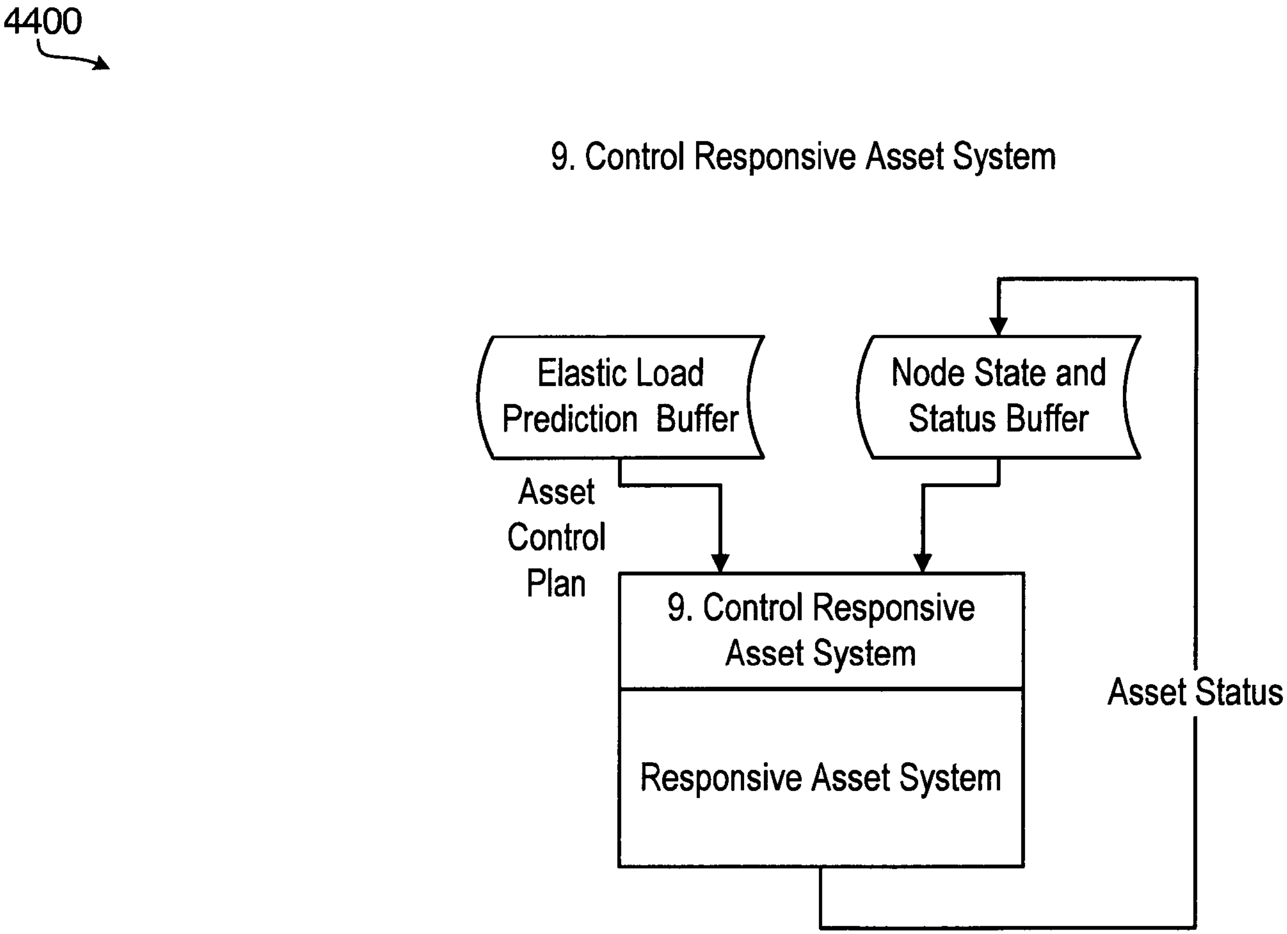

4400
9. Control Responsive Asset System
Elastic Load Prediction Buffer
Node State and Status Buffer
Asset Control Plan
9. Control Responsive Asset System
Responsive Asset System
Asset Status

FIG. 44

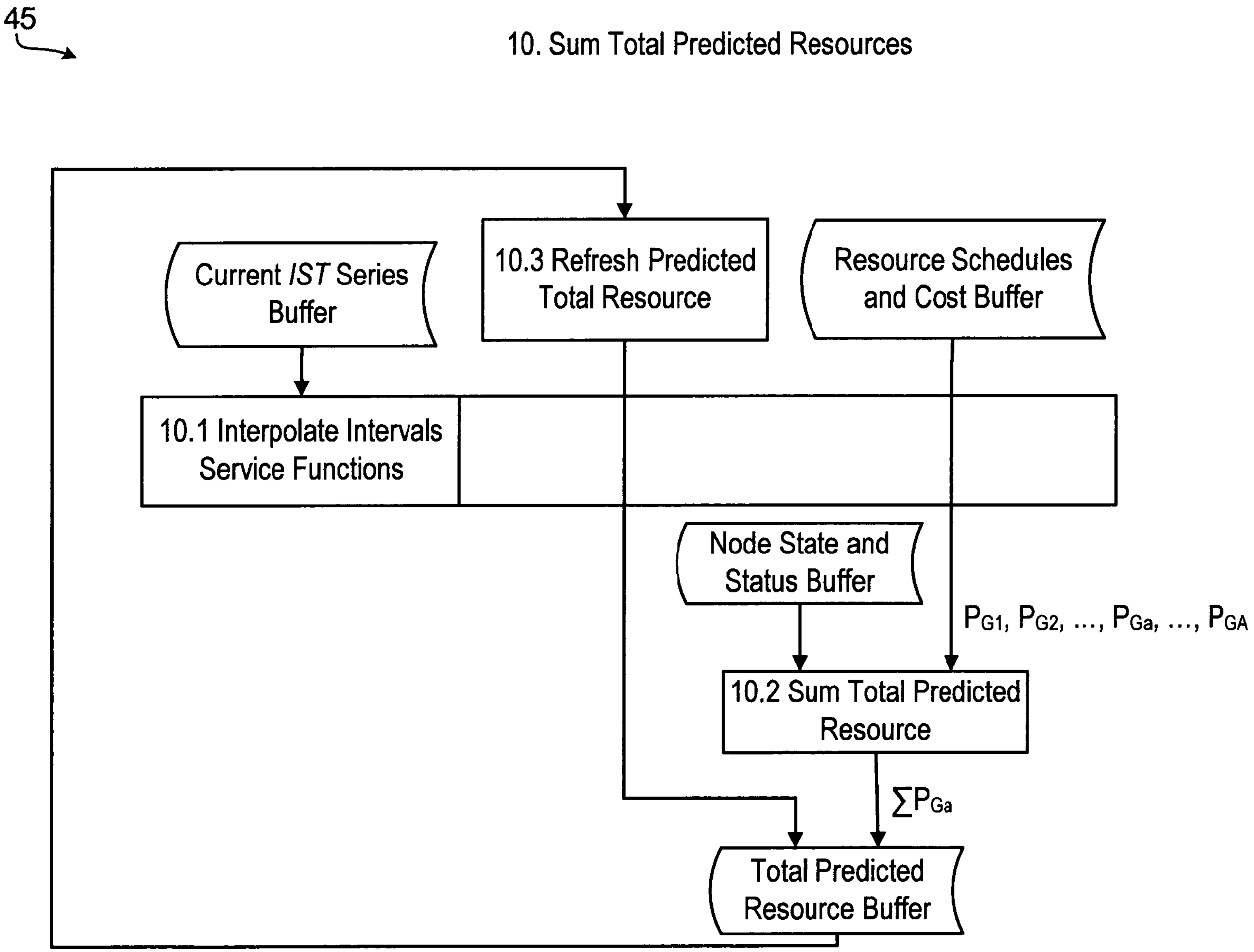

45
10. Sum Total Predicted Resources
Current *IST* Series Buffer
10.3 Refresh Predicted Total Resource
Resource Schedules and Cost Buffer
10.1 Interpolate Intervals Service Functions
Node State and Status Buffer
$P_{G1}, P_{G2}, \ldots, P_{Ga}, \ldots, P_{GA}$
10.2 Sum Total Predicted Resource
$\sum P_{Ga}$
Total Predicted Resource Buffer

FIG. 45

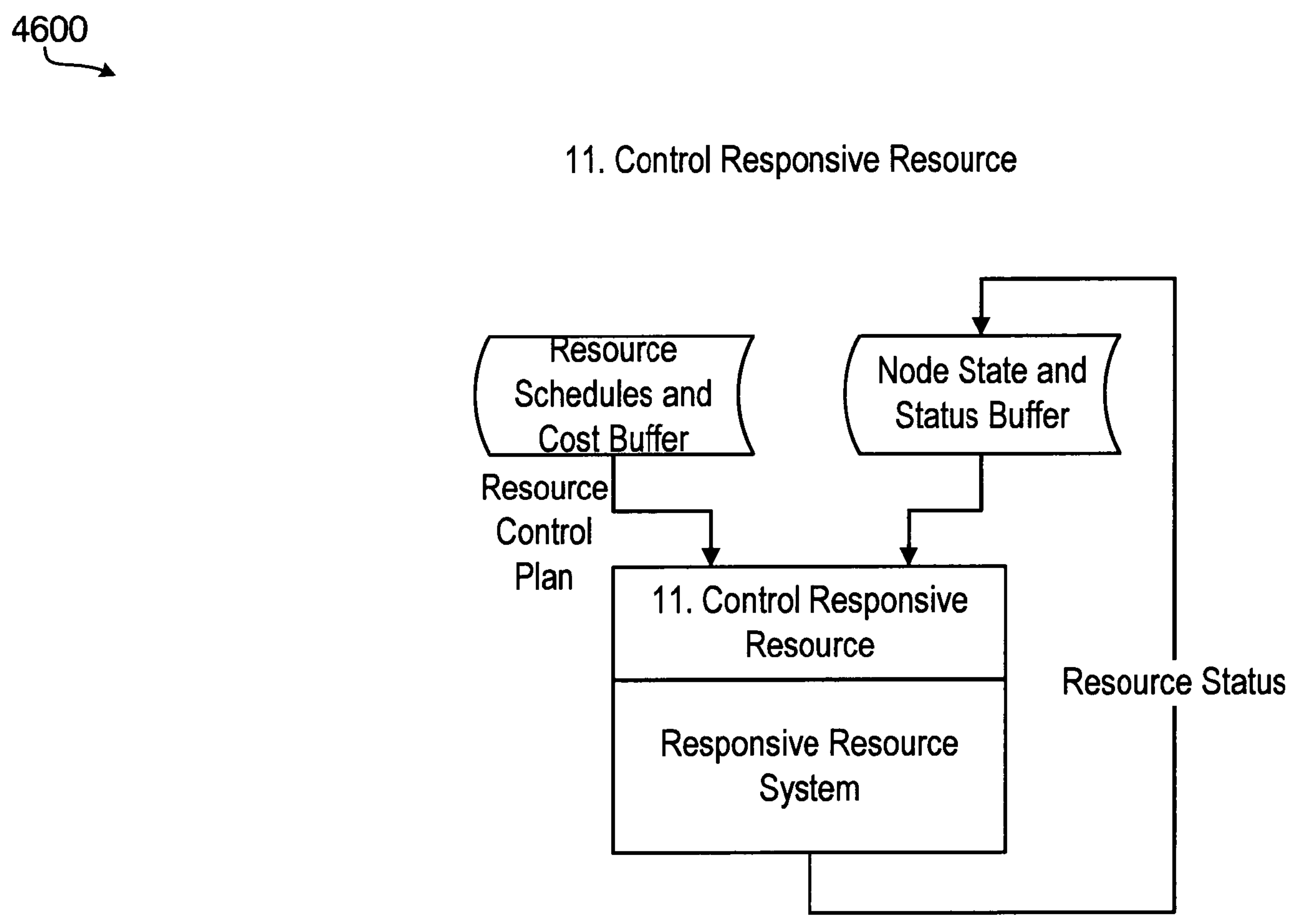

4600
11. Control Responsive Resource
Resource Schedules and Cost Buffer
Resource Control Plan
Node State and Status Buffer
11. Control Responsive Resource
Responsive Resource System
Resource Status

FIG. 46

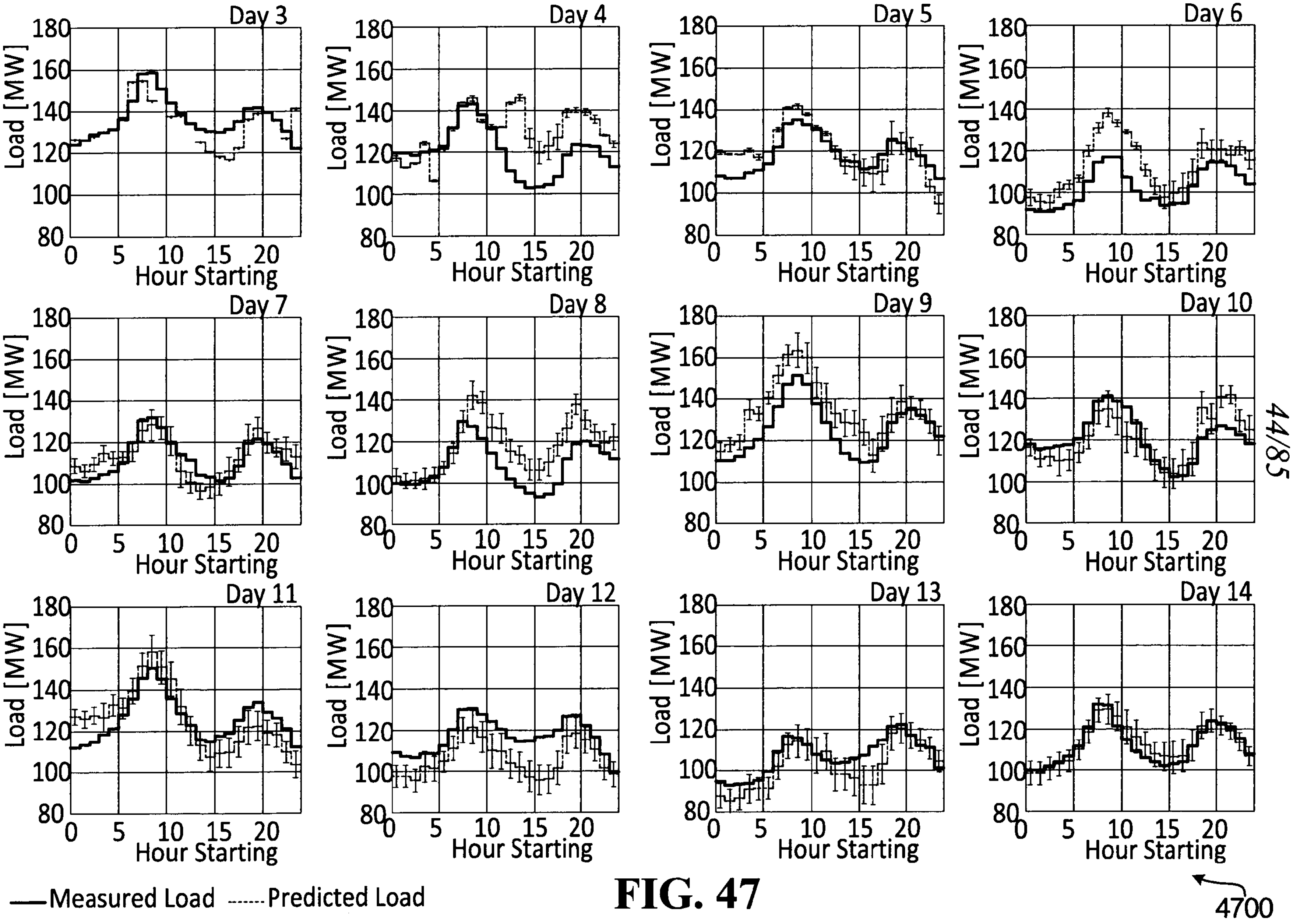

Day 3
Day 4
Day 5
Day 6
Day 7
Day 8
Day 9
Day 10
Day 11
Day 12
Day 13
Day 14
Load [MW]
180
160
140
120
100
80
0
5
10
15
20
Hour Starting
Measured Load
Predicted Load
4700

FIG. 47

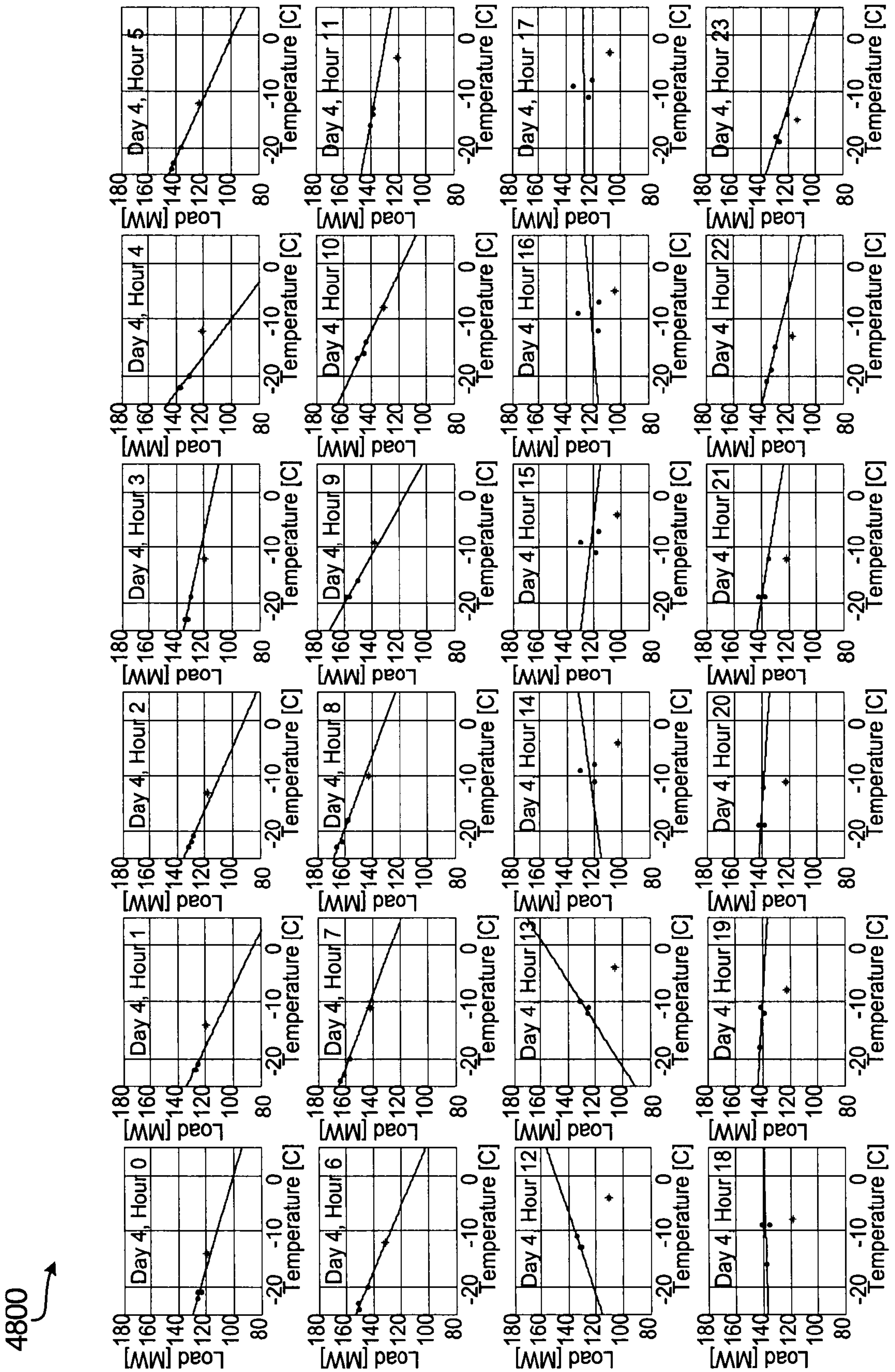

4800
Day 4, Hour 0
Day 4, Hour 1
Day 4, Hour 2
Day 4, Hour 3
Day 4, Hour 4
Day 4, Hour 5
Day 4, Hour 6
Day 4, Hour 7
Day 4, Hour 8
Day 4, Hour 9
Day 4, Hour 10
Day 4, Hour 11
Day 4, Hour 12
Day 4, Hour 13
Day 4, Hour 14
Day 4, Hour 15
Day 4, Hour 16
Day 4, Hour 17
Day 4, Hour 18
Day 4, Hour 19
Day 4, Hour 20
Day 4, Hour 21
Day 4, Hour 22
Day 4, Hour 23
Load [MW]
180
160
140
120
100
80
Temperature [C]
-20
-10
0

FIG. 48

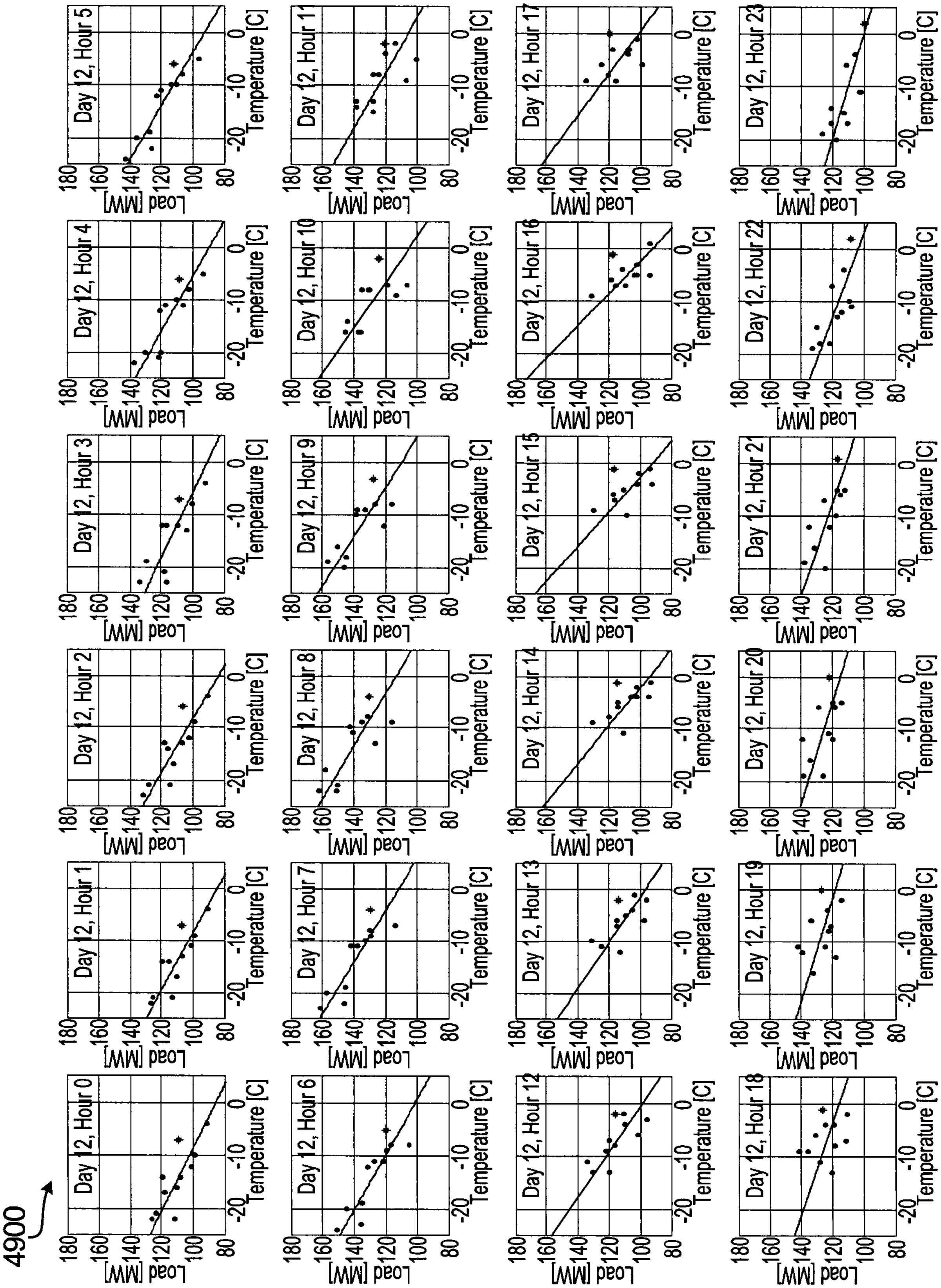

4900
Day 12, Hour 0
Day 12, Hour 1
Day 12, Hour 2
Day 12, Hour 3
Day 12, Hour 4
Day 12, Hour 5
Day 12, Hour 6
Day 12, Hour 7
Day 12, Hour 8
Day 12, Hour 9
Day 12, Hour 10
Day 12, Hour 11
Day 12, Hour 12
Day 12, Hour 13
Day 12, Hour 14
Day 12, Hour 15
Day 12, Hour 16
Day 12, Hour 17
Day 12, Hour 18
Day 12, Hour 19
Day 12, Hour 20
Day 12, Hour 21
Day 12, Hour 22
Day 12, Hour 23
Load [MW]
180
160
140
120
100
80
Temperature [C]
-20
-10
0

FIG. 49

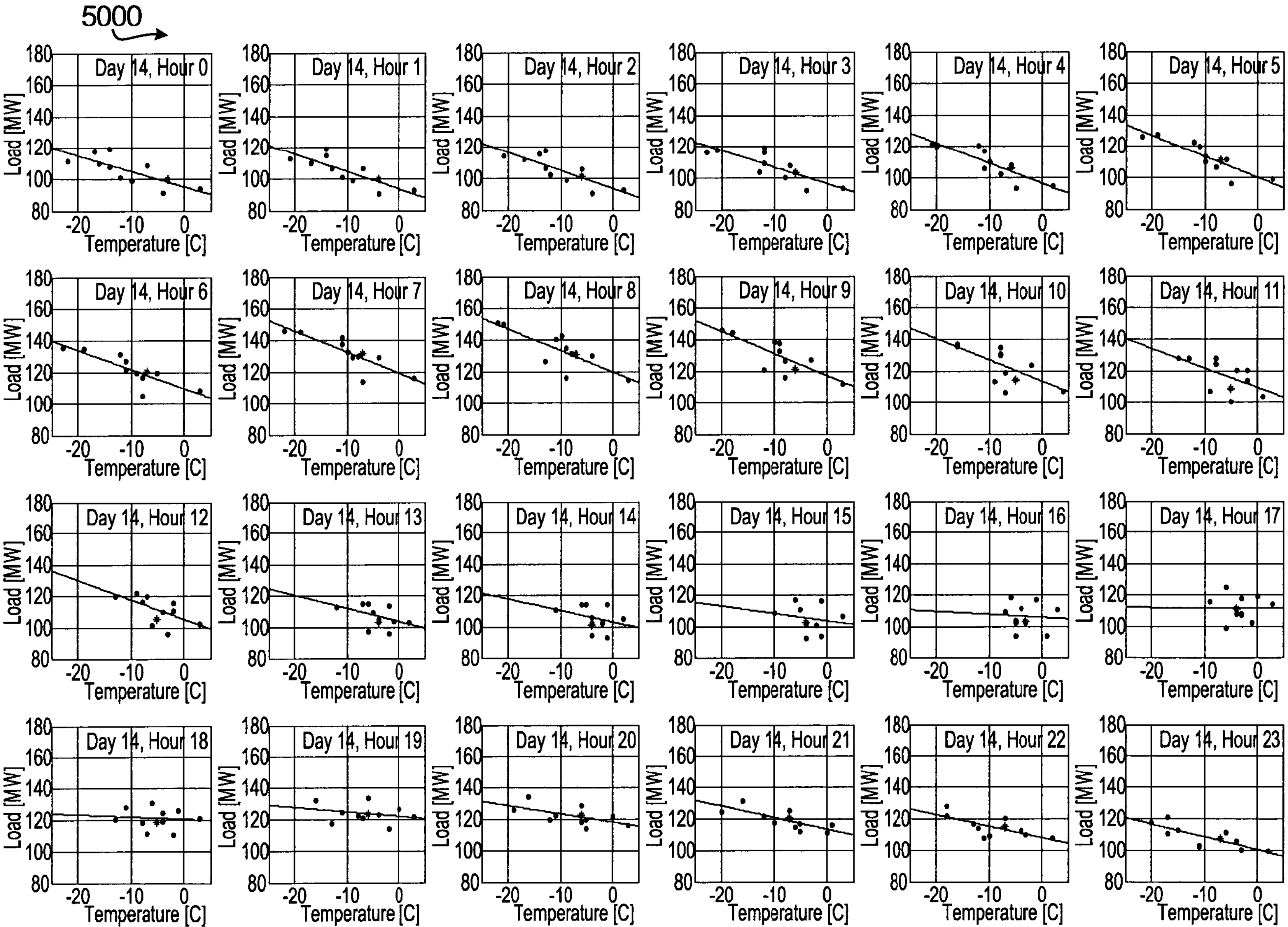

5000
Day 14, Hour 0
Day 14, Hour 1
Day 14, Hour 2
Day 14, Hour 3
Day 14, Hour 4
Day 14, Hour 5
Day 14, Hour 6
Day 14, Hour 7
Day 14, Hour 8
Day 14, Hour 9
Day 14, Hour 10
Day 14, Hour 11
Day 14, Hour 12
Day 14, Hour 13
Day 14, Hour 14
Day 14, Hour 15
Day 14, Hour 16
Day 14, Hour 17
Day 14, Hour 18
Day 14, Hour 19
Day 14, Hour 20
Day 14, Hour 21
Day 14, Hour 22
Day 14, Hour 23
Load [MW]
180
160
140
120
100
80
Temperature [C]
-20
-10
0

FIG. 50

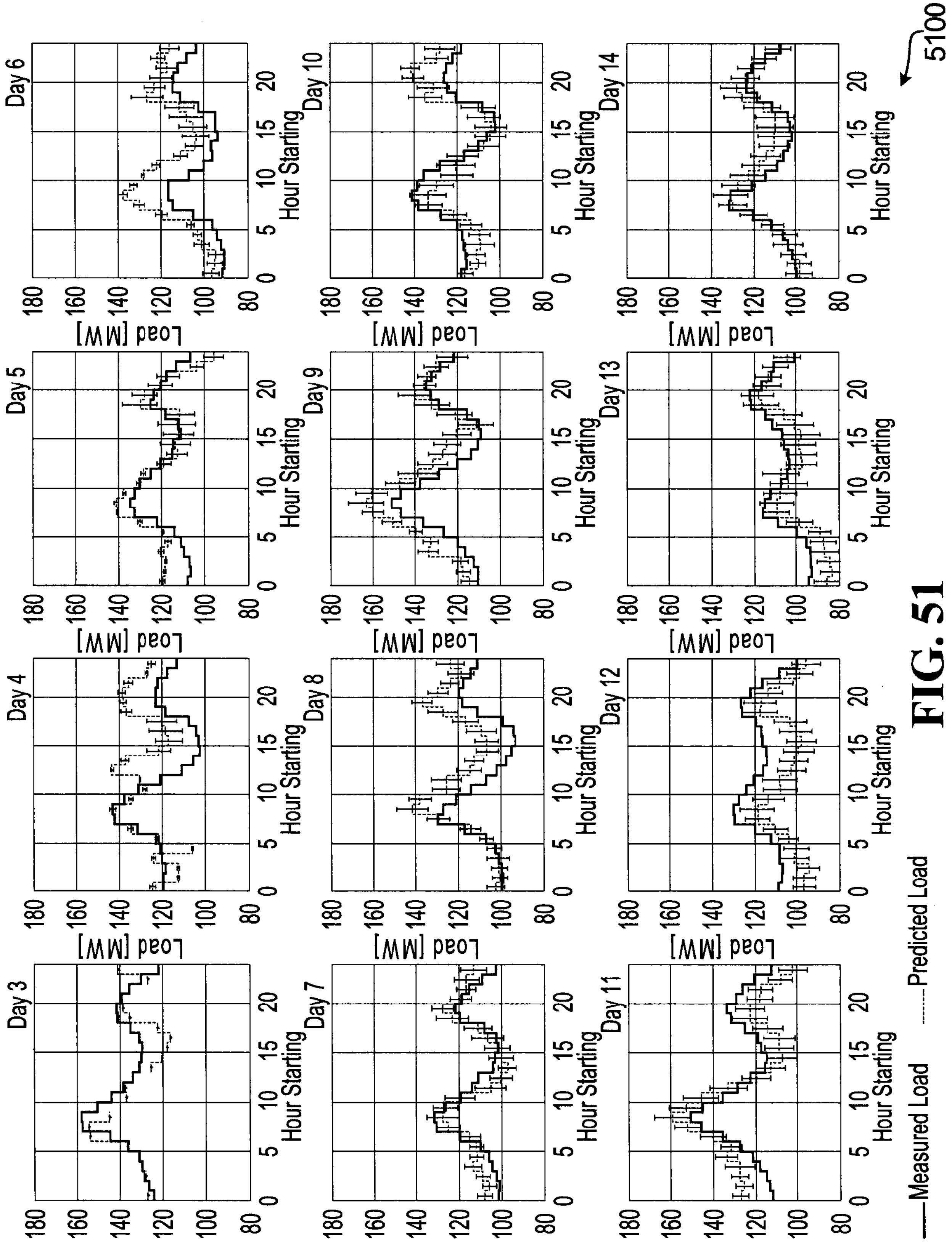

Day 3
Day 4
Day 5
Day 6
Day 7
Day 8
Day 9
Day 10
Day 11
Day 12
Day 13
Day 14
Load [MW]
180
160
140
120
100
80
Hour Starting
0
5
10
15
20
Measured Load
Predicted Load
5100

FIG. 51

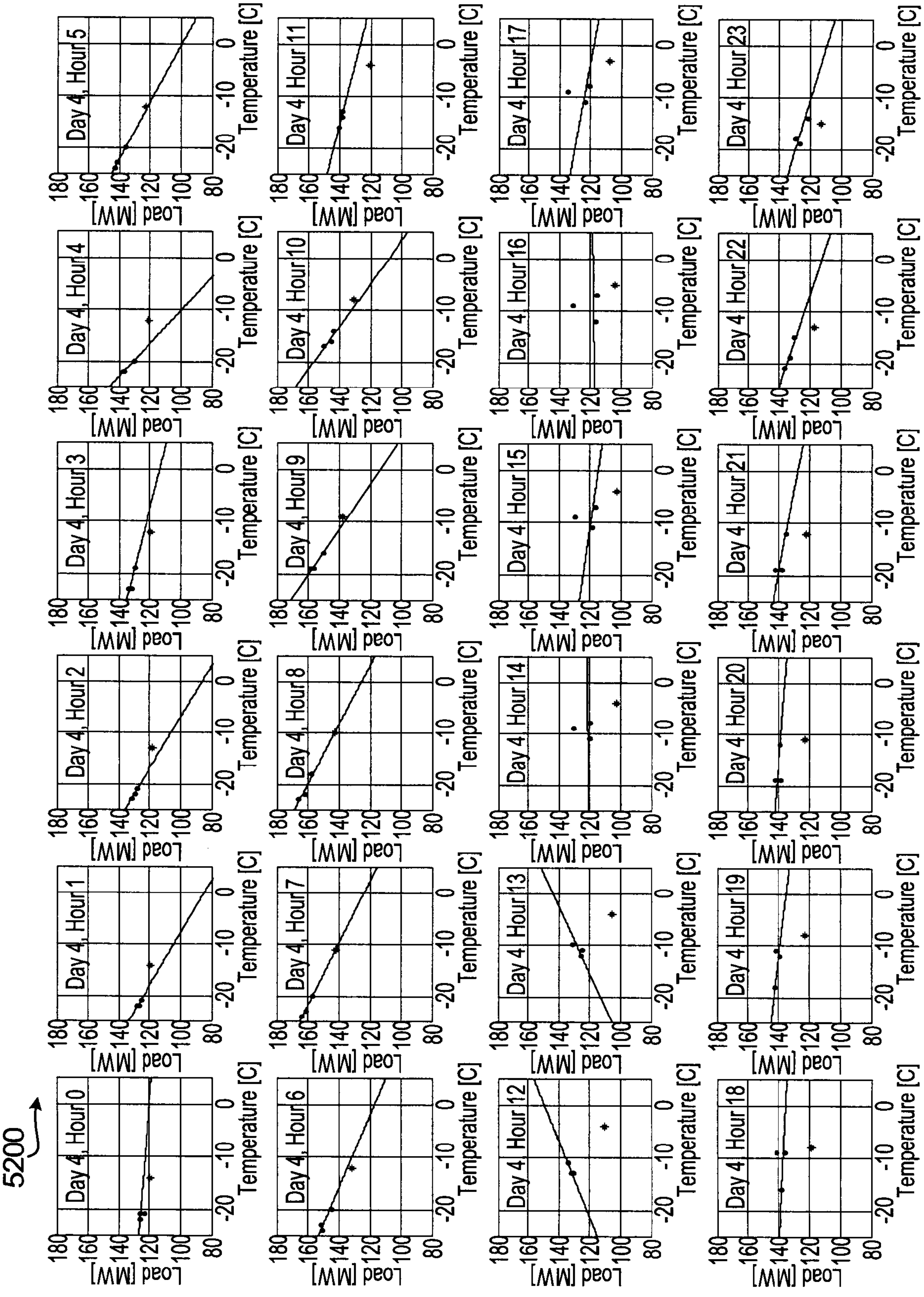

5200
Day 4, Hour 0
Day 4, Hour 1
Day 4, Hour 2
Day 4, Hour 3
Day 4, Hour 4
Day 4, Hour 5
Day 4, Hour 6
Day 4, Hour 7
Day 4, Hour 8
Day 4, Hour 9
Day 4, Hour 10
Day 4, Hour 11
Day 4, Hour 12
Day 4, Hour 13
Day 4, Hour 14
Day 4, Hour 15
Day 4, Hour 16
Day 4, Hour 17
Day 4, Hour 18
Day 4, Hour 19
Day 4, Hour 20
Day 4, Hour 21
Day 4, Hour 22
Day 4, Hour 23
Load [MW]
180
160
140
120
100
80
Temperature [C]
-20
-10
0

FIG. 52

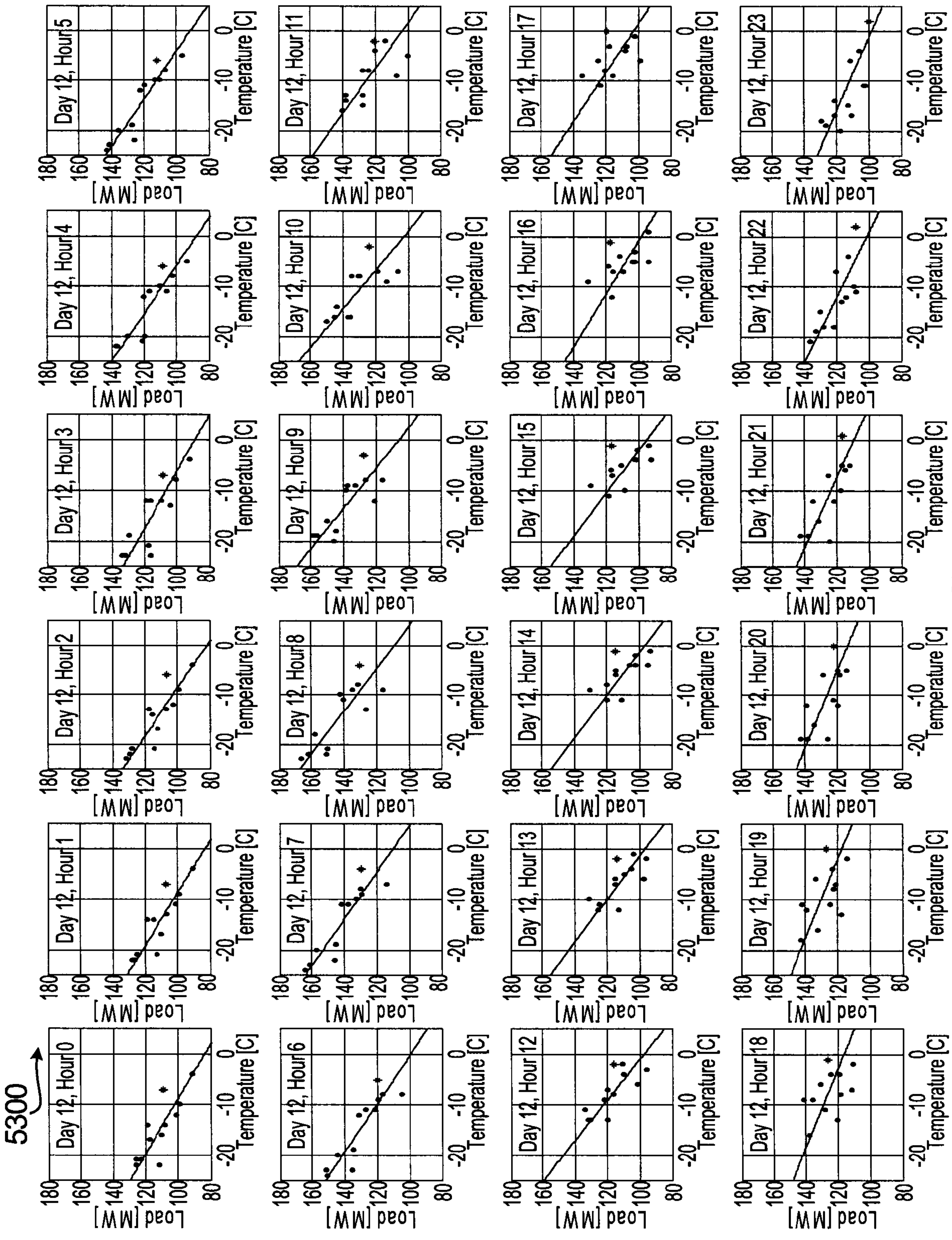

5300
Day 12, Hour 0
Day 12, Hour 1
Day 12, Hour 2
Day 12, Hour 3
Day 12, Hour 4
Day 12, Hour 5
Day 12, Hour 6
Day 12, Hour 7
Day 12, Hour 8
Day 12, Hour 9
Day 12, Hour 10
Day 12, Hour 11
Day 12, Hour 12
Day 12, Hour 13
Day 12, Hour 14
Day 12, Hour 15
Day 12, Hour 16
Day 12, Hour 17
Day 12, Hour 18
Day 12, Hour 19
Day 12, Hour 20
Day 12, Hour 21
Day 12, Hour 22
Day 12, Hour 23
Load [MW]
180
160
140
120
100
80
Temperature [C]
-20
-10
0

FIG. 53

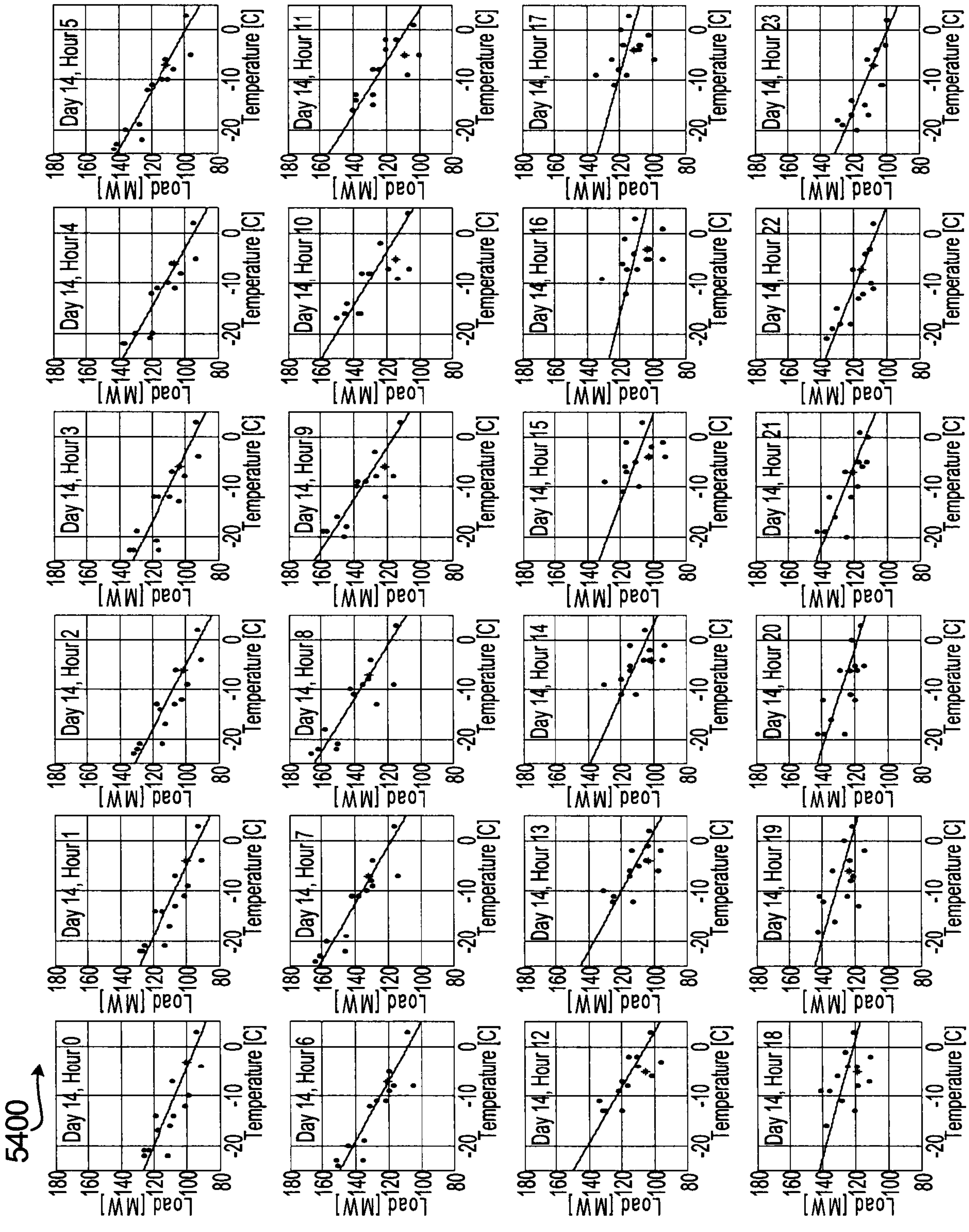

5400
Day 14, Hour 0
Day 14, Hour 1
Day 14, Hour 2
Day 14, Hour 3
Day 14, Hour 4
Day 14, Hour 5
Day 14, Hour 6
Day 14, Hour 7
Day 14, Hour 8
Day 14, Hour 9
Day 14, Hour 10
Day 14, Hour 11
Day 14, Hour 12
Day 14, Hour 13
Day 14, Hour 14
Day 14, Hour 15
Day 14, Hour 16
Day 14, Hour 17
Day 14, Hour 18
Day 14, Hour 19
Day 14, Hour 20
Day 14, Hour 21
Day 14, Hour 22
Day 14, Hour 23
Load [MW]
180
160
140
120
100
80
Temperature [C]
-20
-10
0

FIG. 54

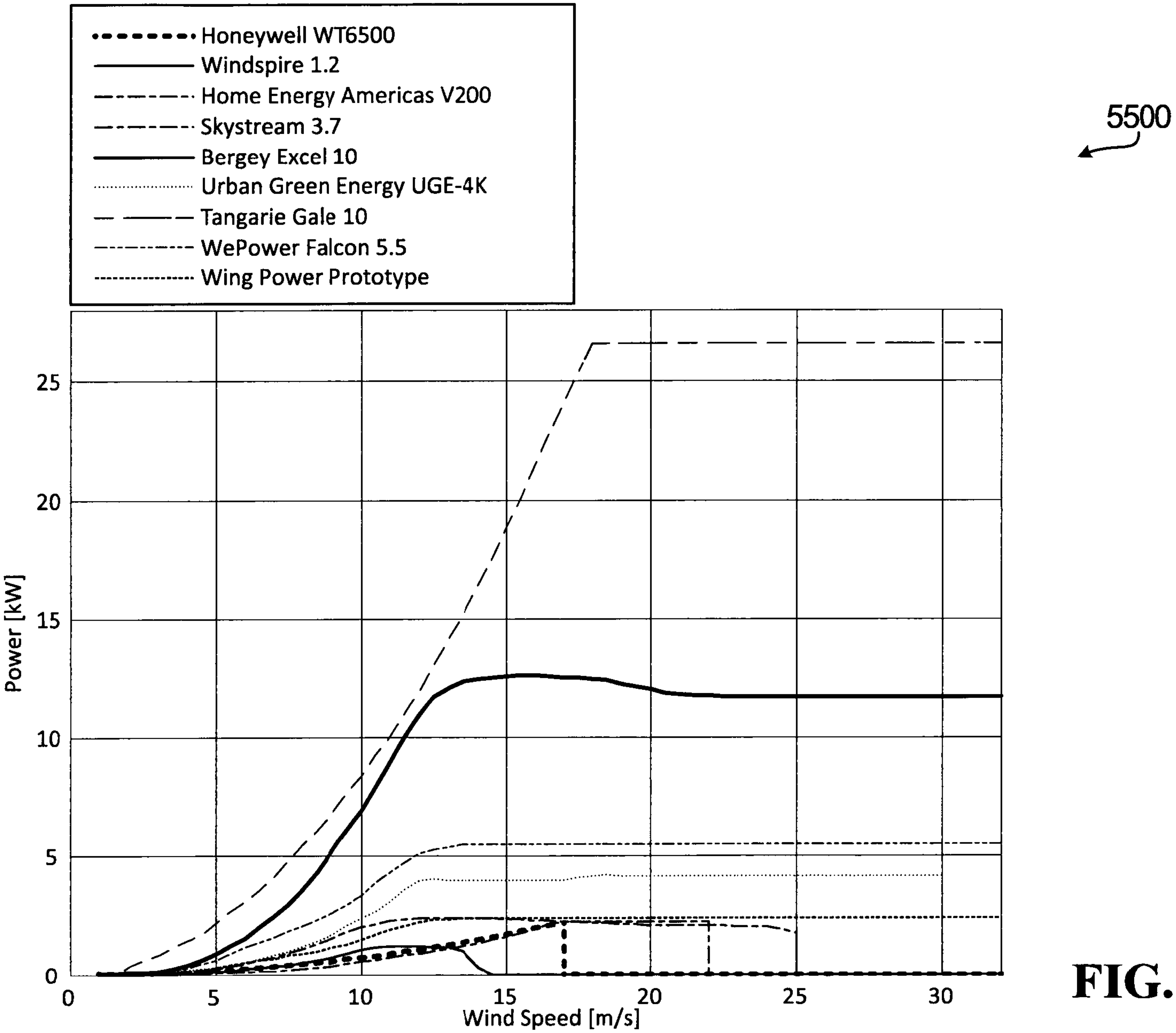

5500
Honeywell WT6500
Windspire 1.2
Home Energy Americas V200
Skystream 3.7
Bergey Excel 10
Urban Green Energy UGE-4K
Tangarie Gale 10
WePower Falcon 5.5
Wing Power Prototype
Power [kW]
0
5
10
15
20
25
Wind Speed [m/s]
0
5
10
15
20
25
30

FIG. 55

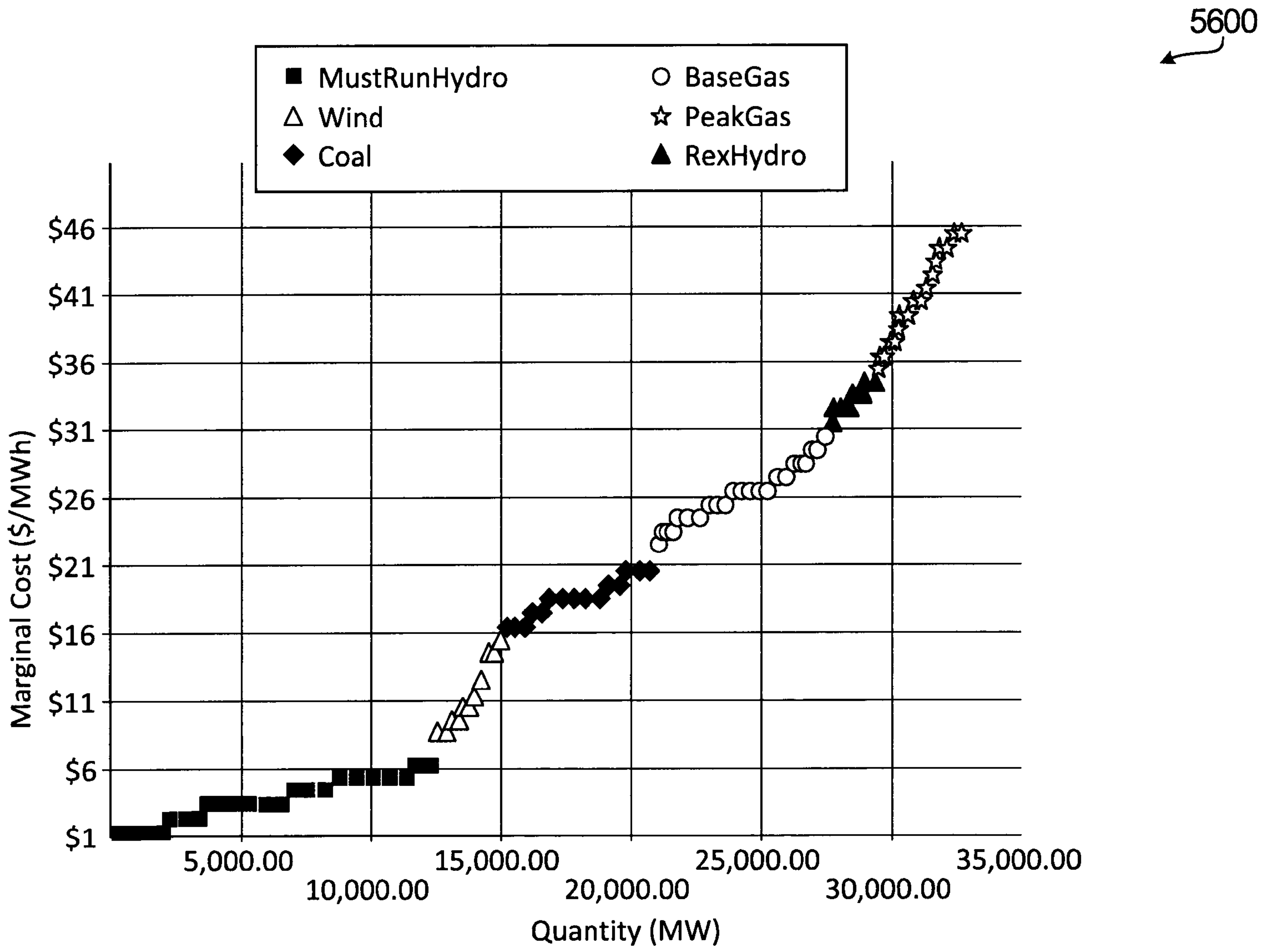
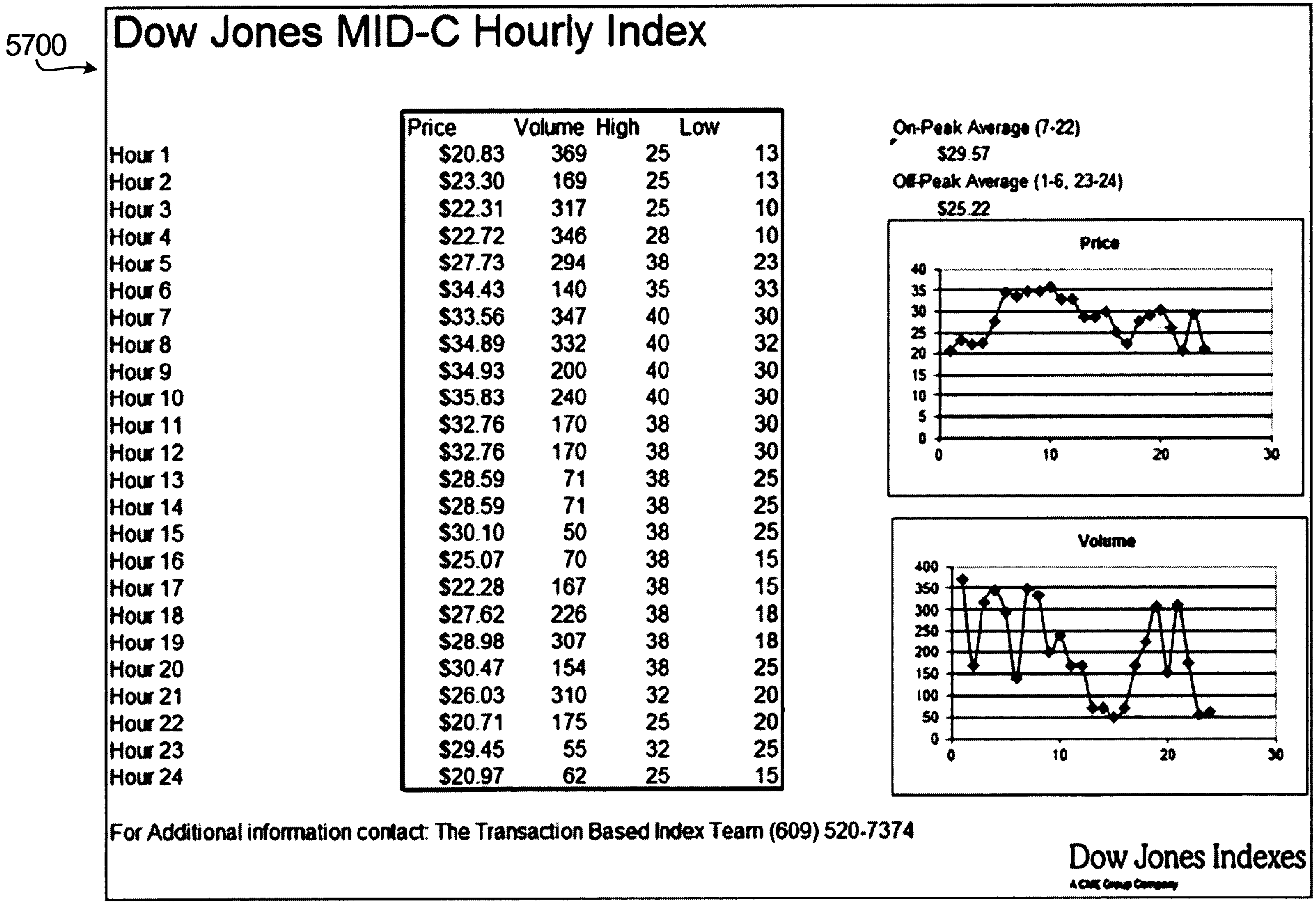

5600
■ MustRunHydro
△ Wind
◆ Coal
○ BaseGas
☆ PeakGas
▲ RexHydro
Marginal Cost ($/MWh)
$46
$41
$36
$31
$26
$21
$16
$11
$6
$1
5,000.00
10,000.00
15,000.00
20,000.00
25,000.00
30,000.00
35,000.00
Quantity (MW)

FIG. 56



Dow Jones MID-C Hourly Index

	Price	Volume	High	Low
Hour 1	$20.83	369	25	13
Hour 2	$23.30	169	25	13
Hour 3	$22.31	317	25	10
Hour 4	$22.72	346	28	10
Hour 5	$27.73	294	38	23
Hour 6	$34.43	140	35	33
Hour 7	$33.56	347	40	30
Hour 8	$34.89	332	40	32
Hour 9	$34.93	200	40	30
Hour 10	$35.83	240	40	30
Hour 11	$32.76	170	38	30
Hour 12	$32.76	170	38	30
Hour 13	$28.59	71	38	25
Hour 14	$28.59	71	38	25
Hour 15	$30.10	50	38	25
Hour 16	$25.07	70	38	15
Hour 17	$22.28	167	38	15
Hour 18	$27.62	226	38	18
Hour 19	$28.98	307	38	18
Hour 20	$30.47	154	38	25
Hour 21	$26.03	310	32	20
Hour 22	$20.71	175	25	20
Hour 23	$29.45	55	32	25
Hour 24	$20.97	62	25	15

On-Peak Average (7-22)
$29.57
Off-Peak Average (1-6, 23-24)
$25.22

For Additional information contact: The Transaction Based Index Team (609) 520-7374

Dow Jones Indexes
A CME Group Company

FIG. 57

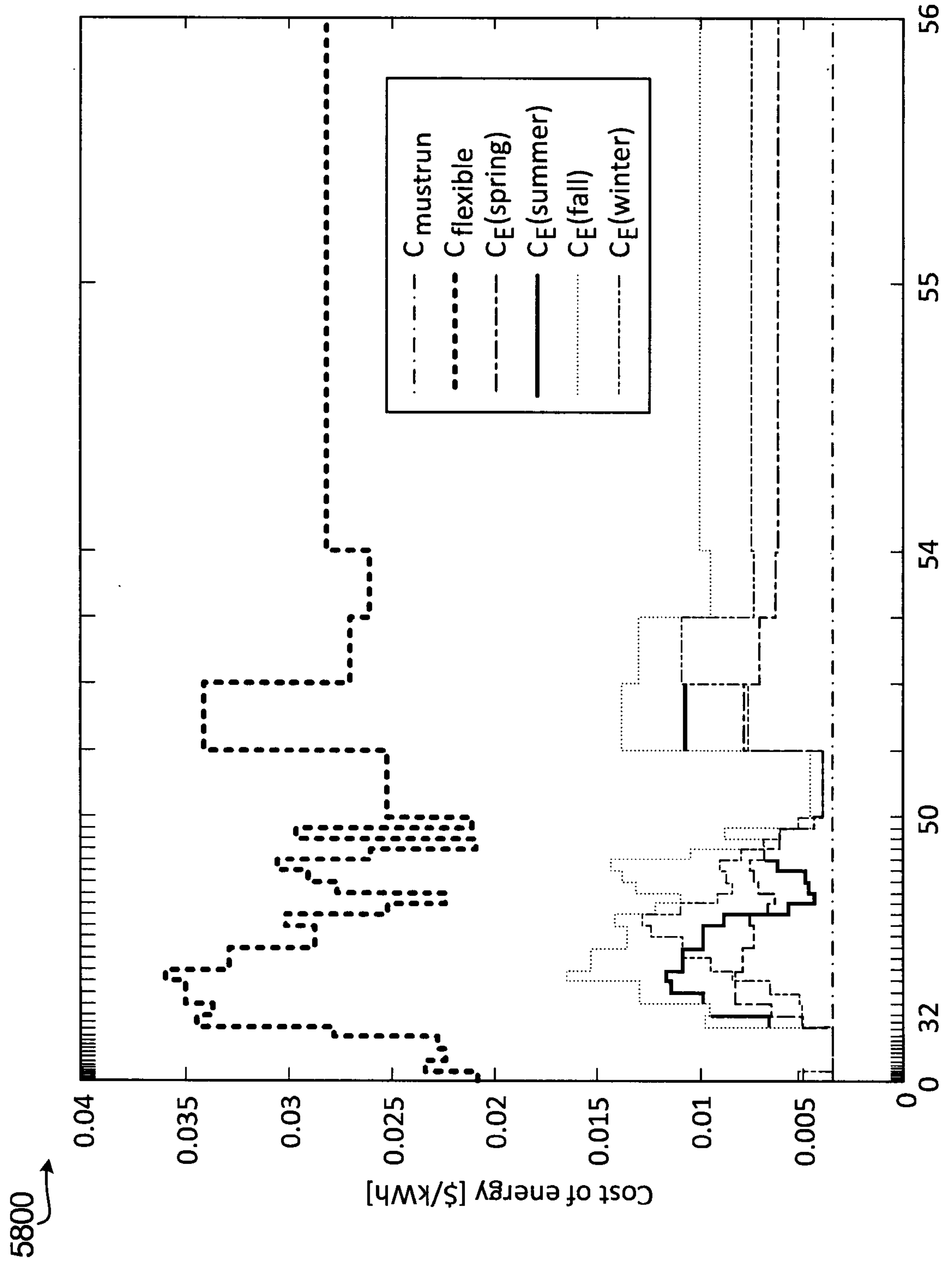

5800
Cost of energy [$/kWh]
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
n
0
32
50
54
55
56
$C_{mustrun}$
$C_{flexible}$
C_E(spring)
C_E(summer)
C_E(fall)
C_E(winter)

FIG. 58

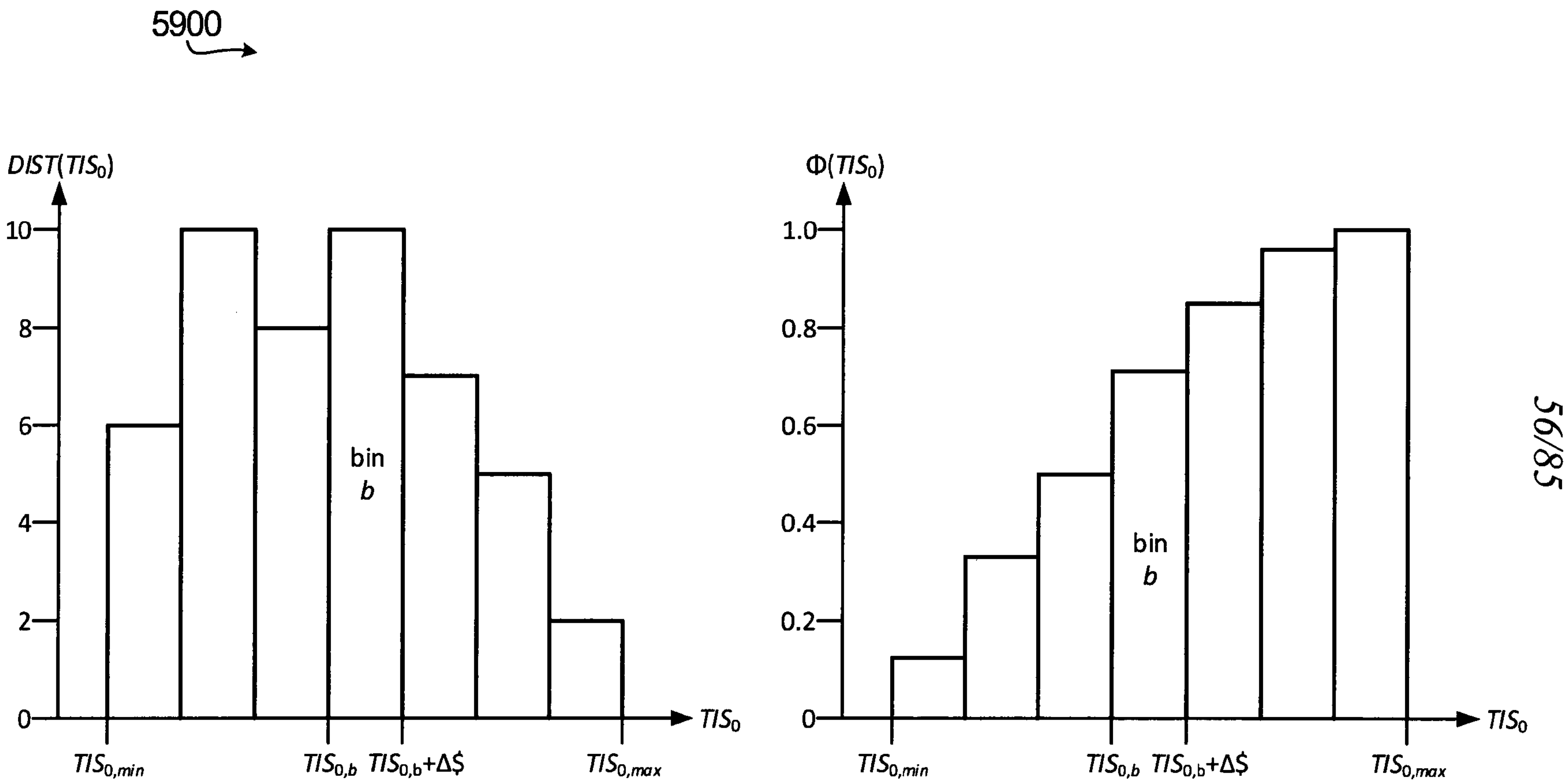

5900
DIST(TIS0)
10
8
6
4
2
0
bin
b
TIS0
TIS0,min
TIS0,b
TIS0,b+Δ$
TIS0,max
Φ(TIS0)
1.0
0.8
0.6
0.4
0.2
0
bin
b
TIS0
TIS0,min
TIS0,b
TIS0,b+Δ$
TIS0,max

FIG. 59

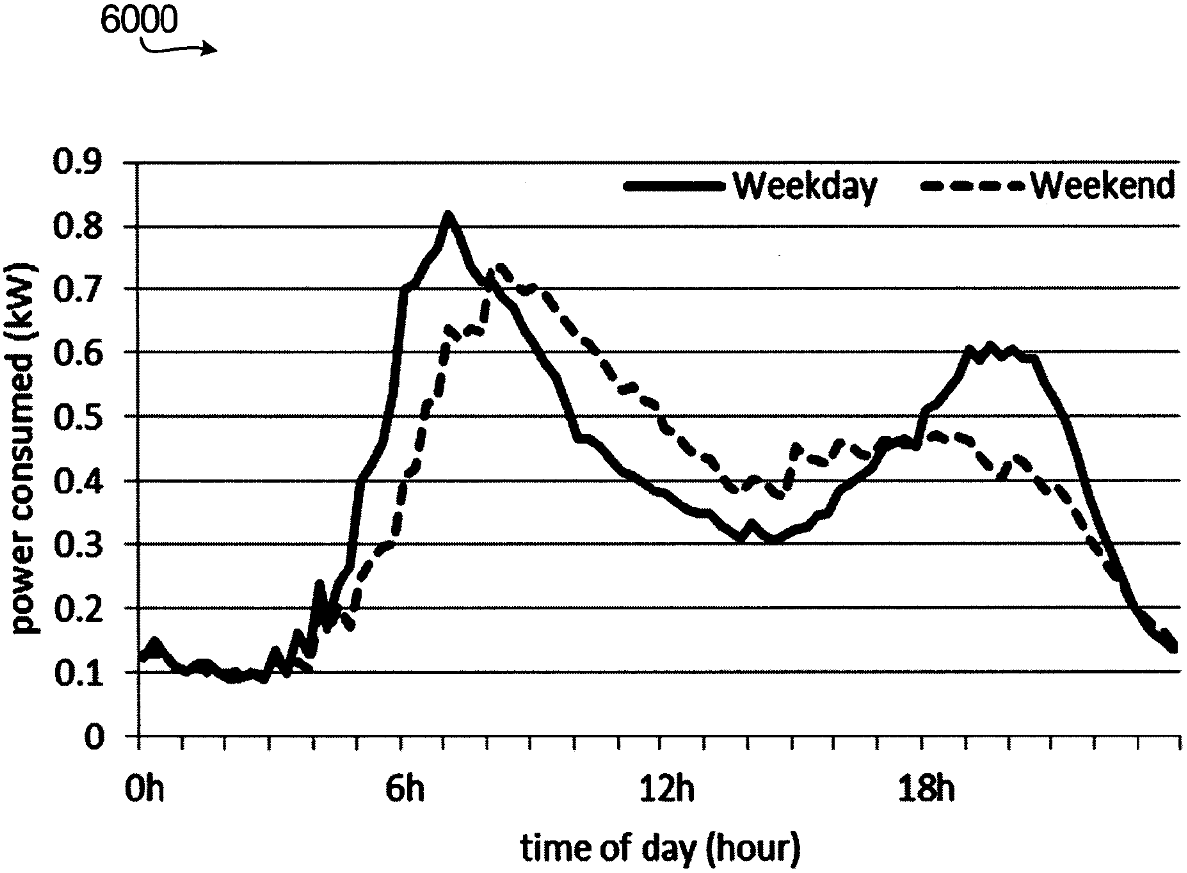

6000
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
power consumed (kW)
Weekday
Weekend
0h
6h
12h
18h
time of day (hour)

FIG. 60

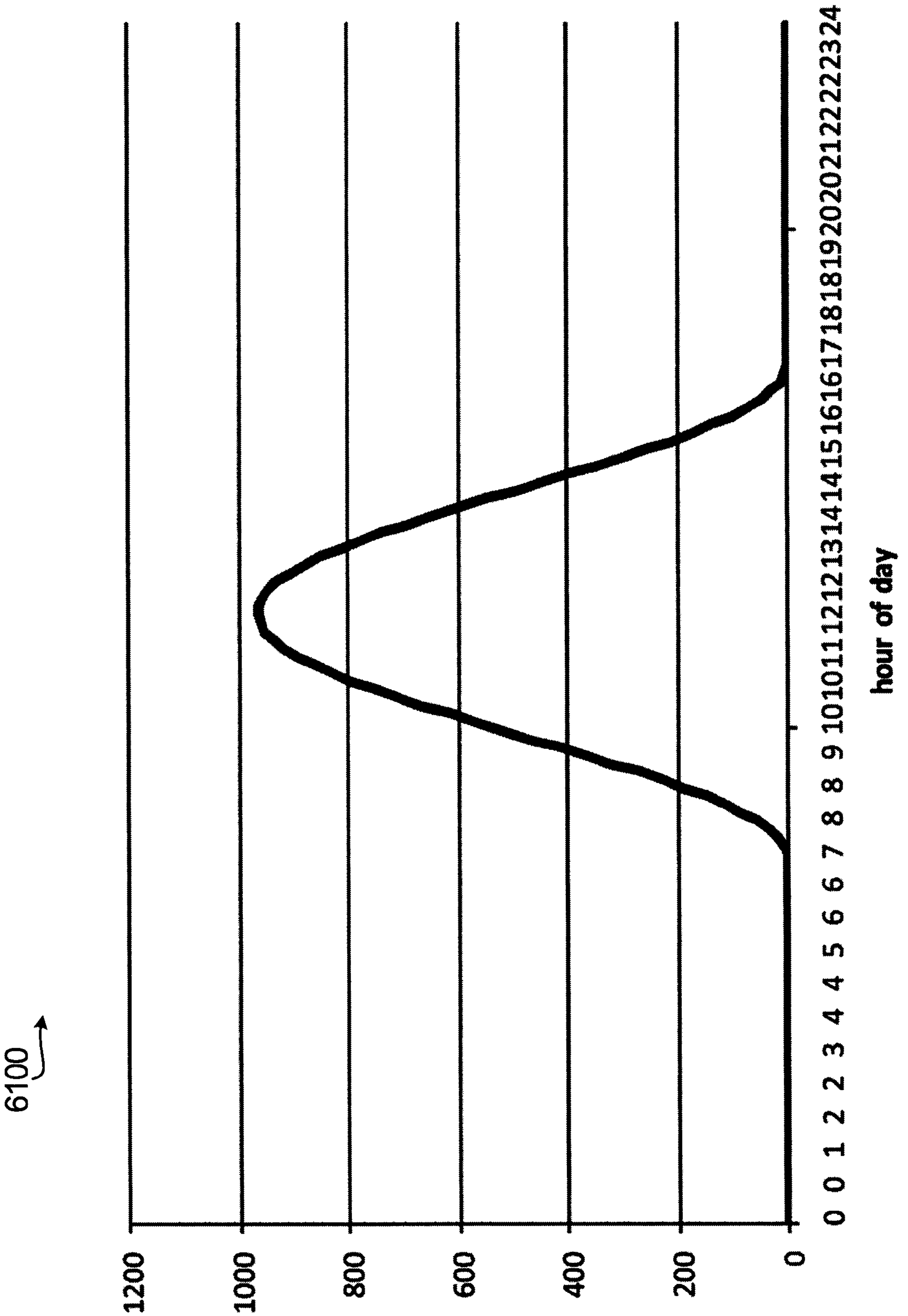

6100
solar power density (W/m^2)
1200
1000
800
600
400
200
0
0 0 1 2 2 3 4 4 5 6 6 7 8 8 9 10 10 11 12 12 13 14 14 15 16 16 17 18 18 19 20 20 21 22 22 23 24
hour of day

FIG. 61

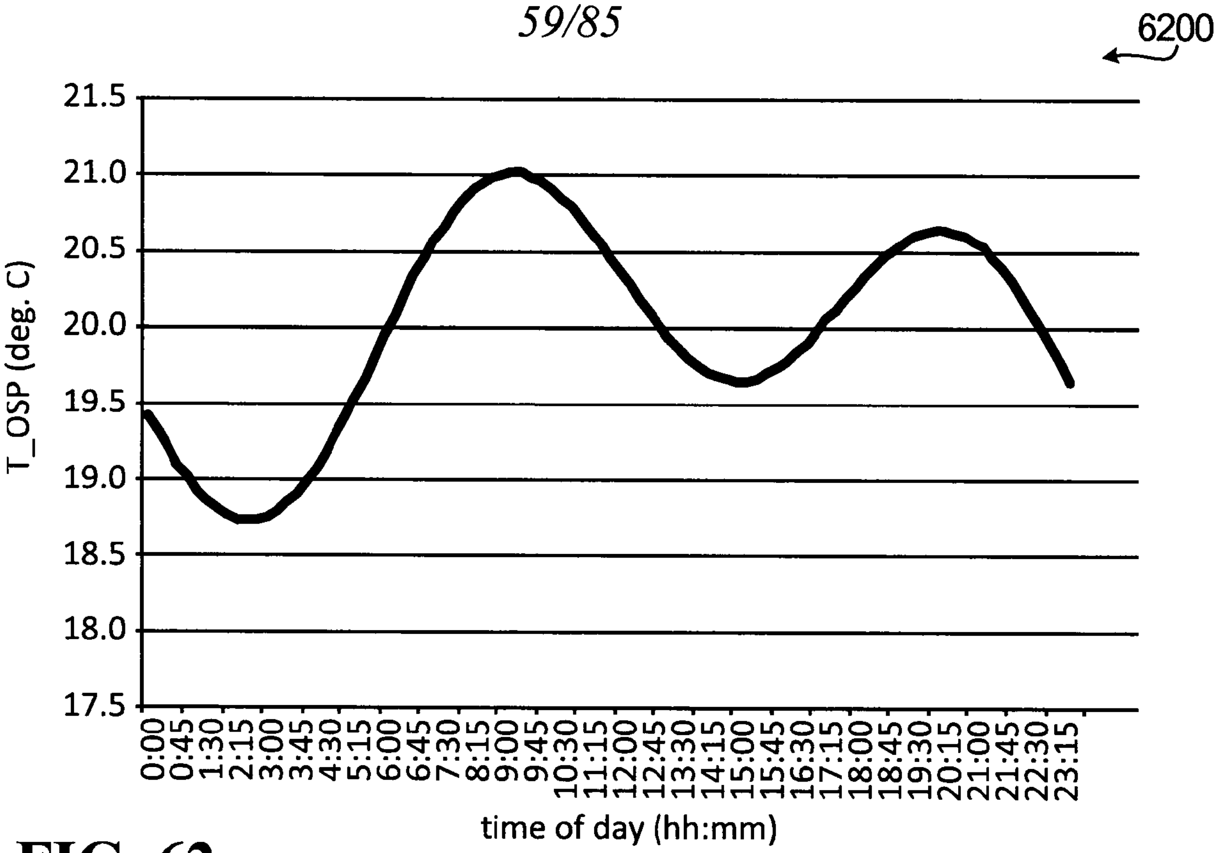

6200
T_OSP (deg. C)
21.5
21.0
20.5
20.0
19.5
19.0
18.5
18.0
17.5
0:00
0:45
1:30
2:15
3:00
3:45
4:30
5:15
6:00
6:45
7:30
8:15
9:00
9:45
10:30
11:15
12:00
12:45
13:30
14:15
15:00
15:45
16:30
17:15
18:00
18:45
19:30
20:15
21:00
21:45
22:30
23:15
time of day (hh:mm)

FIG. 62

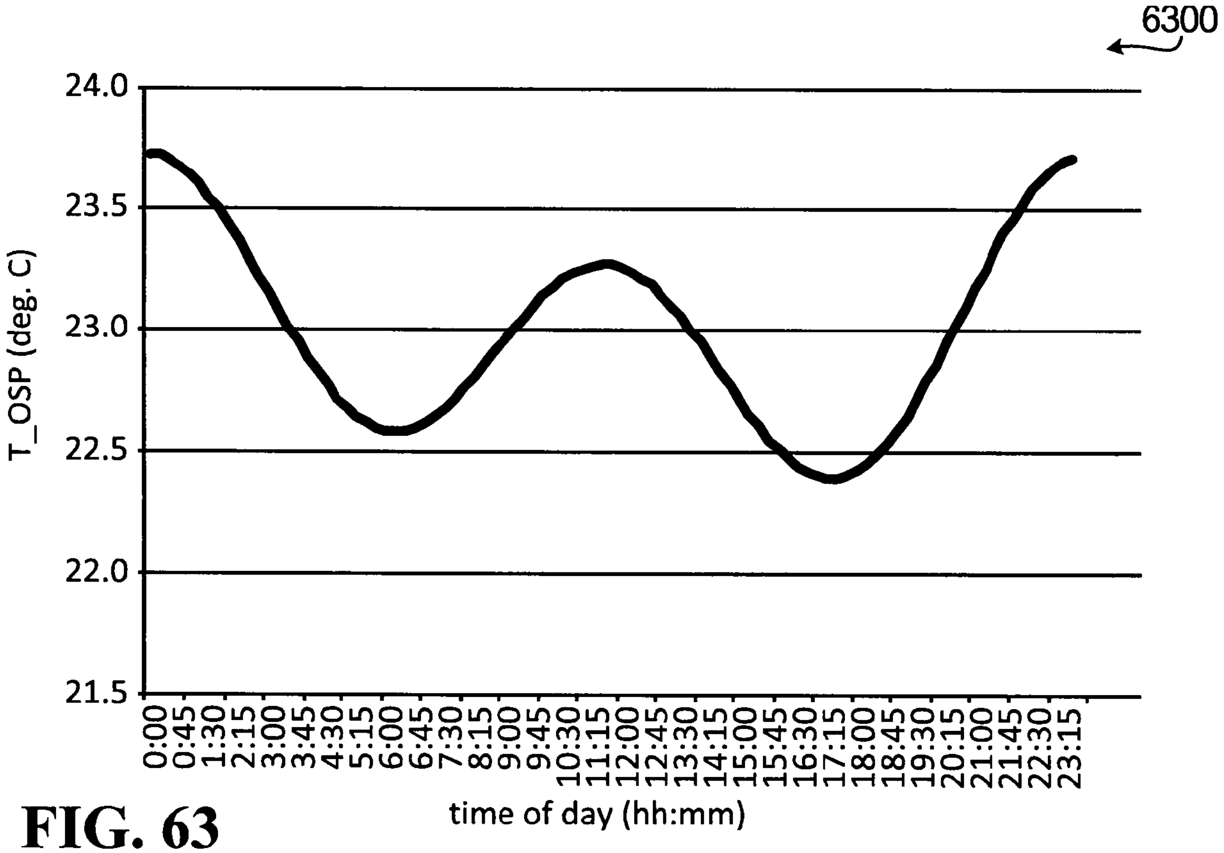

6300
T_OSP (deg. C)
24.0
23.5
23.0
22.5
22.0
21.5
0:00
0:45
1:30
2:15
3:00
3:45
4:30
5:15
6:00
6:45
7:30
8:15
9:00
9:45
10:30
11:15
12:00
12:45
13:30
14:15
15:00
15:45
16:30
17:15
18:00
18:45
19:30
20:15
21:00
21:45
22:30
23:15
time of day (hh:mm)

FIG. 63

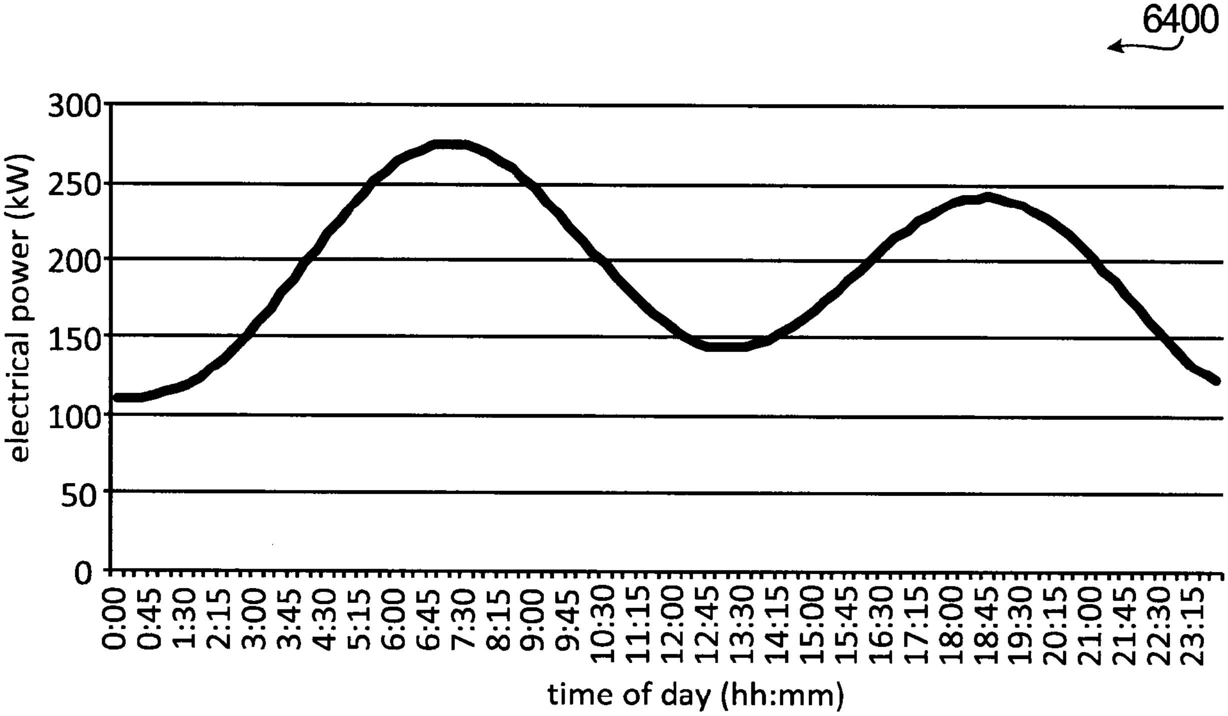

6400
300
250
200
150
100
50
0
electrical power (kW)
0:00
0:45
1:30
2:15
3:00
3:45
4:30
5:15
6:00
6:45
7:30
8:15
9:00
9:45
10:30
11:15
12:00
12:45
13:30
14:15
15:00
15:45
16:30
17:15
18:00
18:45
19:30
20:15
21:00
21:45
22:30
23:15
time of day (hh:mm)

FIG. 64

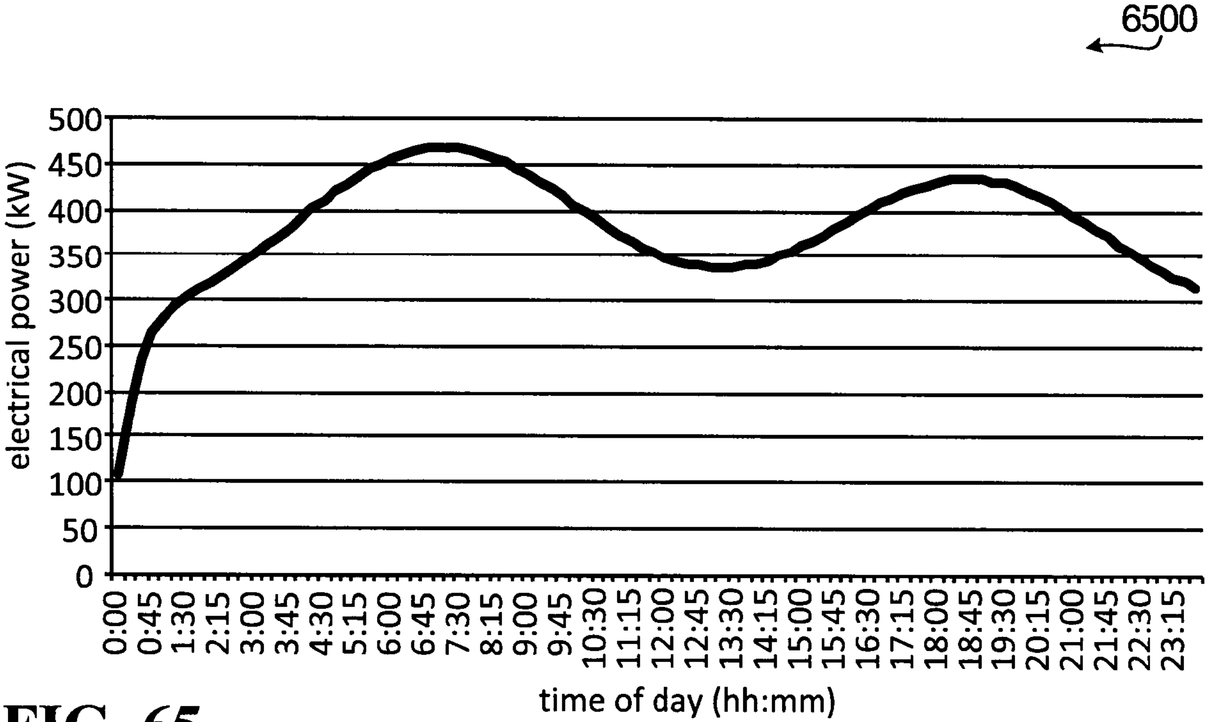

6500
500
450
400
350
300
250
200
150
100
50
0
electrical power (kW)
0:00
0:45
1:30
2:15
3:00
3:45
4:30
5:15
6:00
6:45
7:30
8:15
9:00
9:45
10:30
11:15
12:00
12:45
13:30
14:15
15:00
15:45
16:30
17:15
18:00
18:45
19:30
20:15
21:00
21:45
22:30
23:15
time of day (hh:mm)

FIG. 65

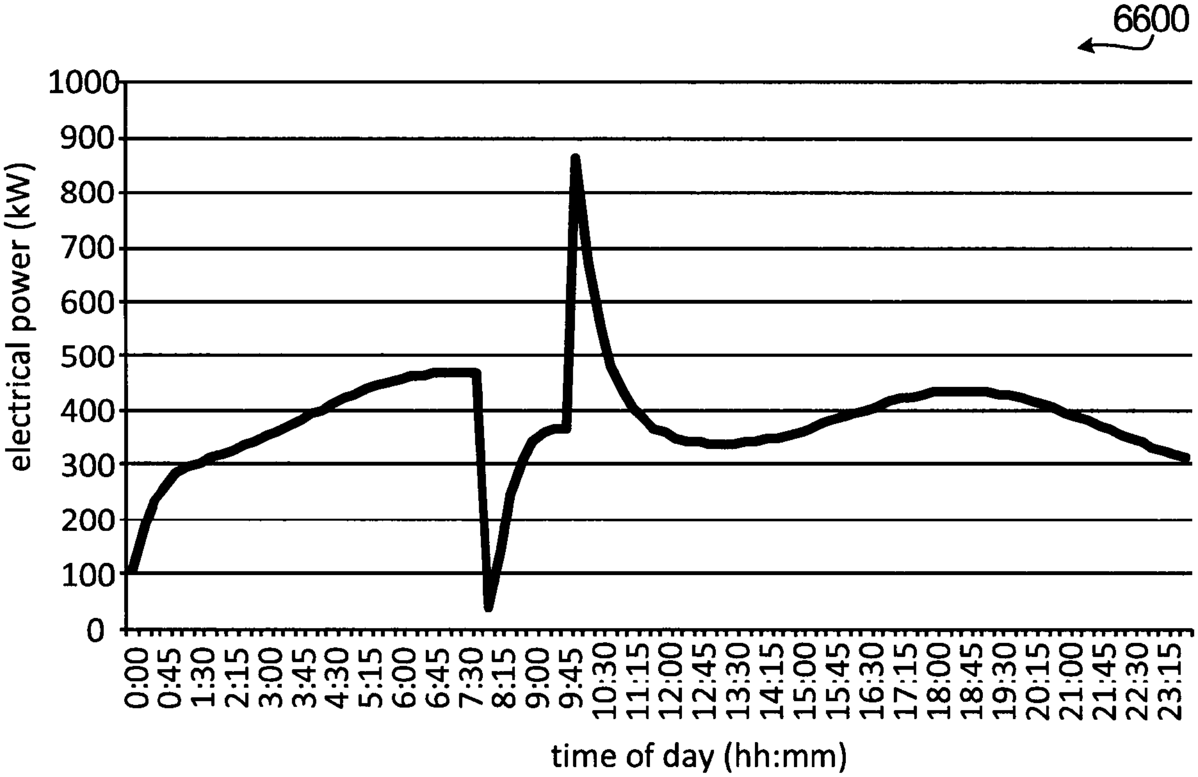

6600
1000
900
800
700
600
500
400
300
200
100
0
electrical power (kW)
0:00
0:45
1:30
2:15
3:00
3:45
4:30
5:15
6:00
6:45
7:30
8:15
9:00
9:45
10:30
11:15
12:00
12:45
13:30
14:15
15:00
15:45
16:30
17:15
18:00
18:45
19:30
20:15
21:00
21:45
22:30
23:15
time of day (hh:mm)

FIG. 66

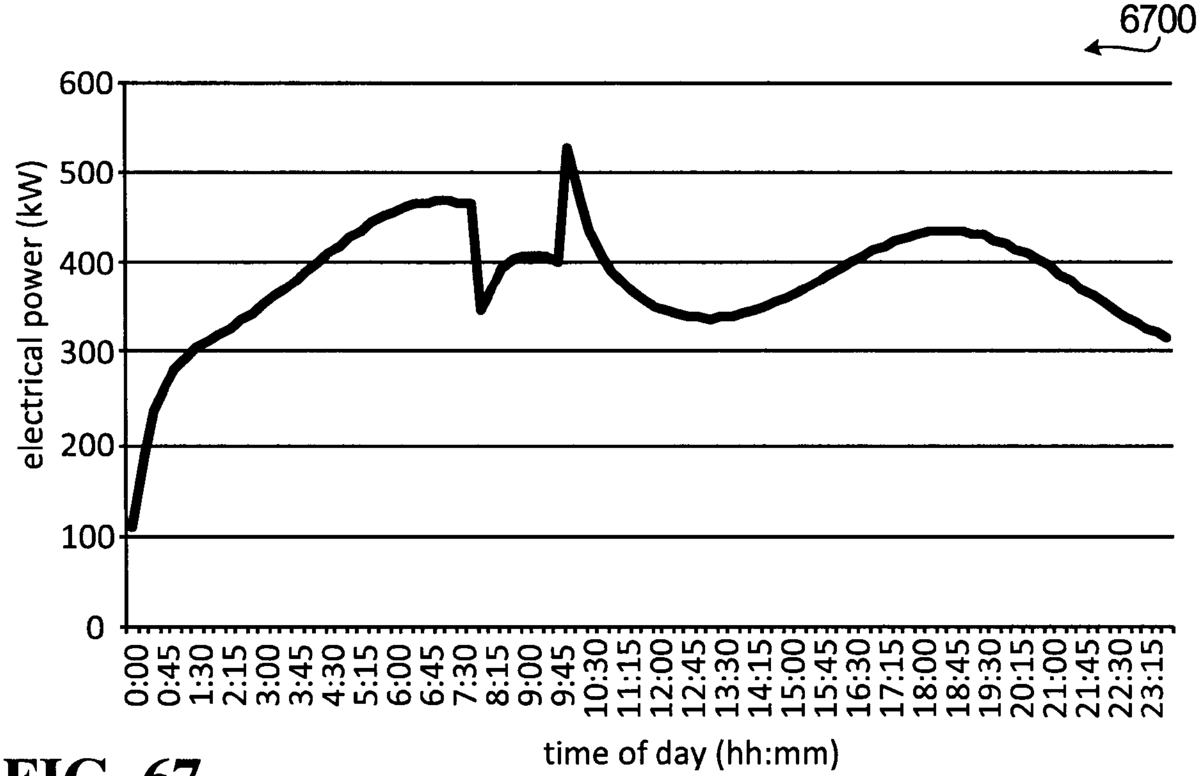

6700
600
500
400
300
200
100
0
electrical power (kW)
0:00
0:45
1:30
2:15
3:00
3:45
4:30
5:15
6:00
6:45
7:30
8:15
9:00
9:45
10:30
11:15
12:00
12:45
13:30
14:15
15:00
15:45
16:30
17:15
18:00
18:45
19:30
20:15
21:00
21:45
22:30
23:15
time of day (hh:mm)

FIG. 67

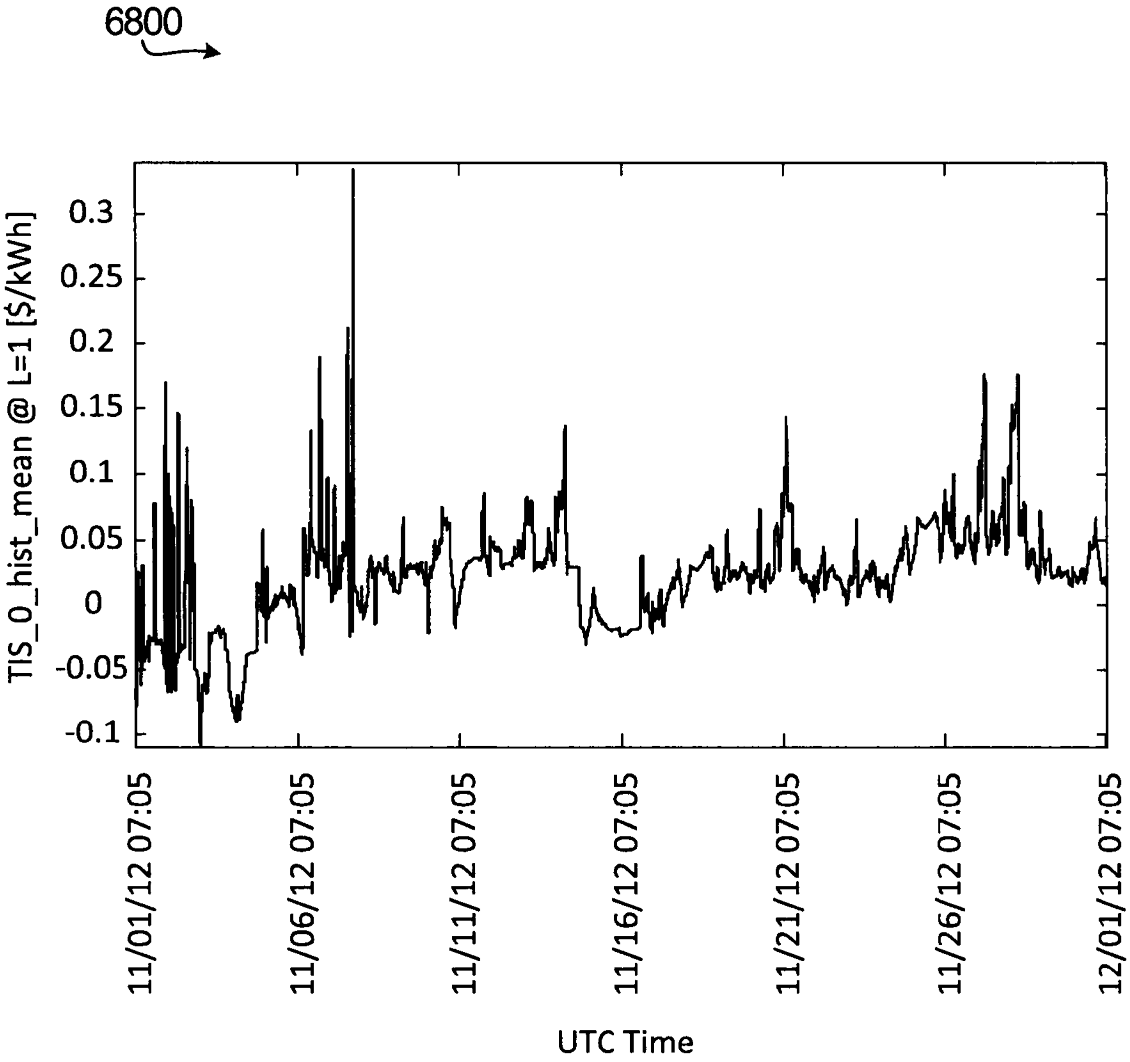

6800
TIS_0_hist_mean @ L=1 [$/kWh]
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
11/01/12 07:05
11/06/12 07:05
11/11/12 07:05
11/16/12 07:05
11/21/12 07:05
11/26/12 07:05
12/01/12 07:05
UTC Time

FIG. 68

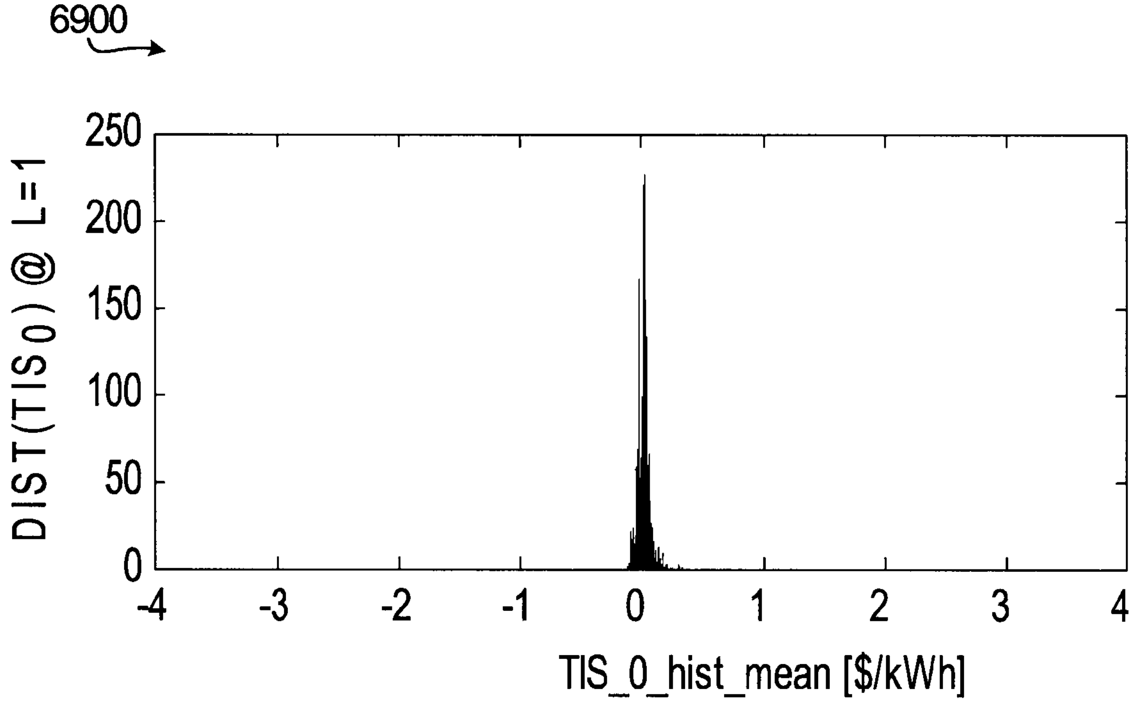

6900
250
200
150
100
50
0
-4
-3
-2
-1
0
1
2
3
4
DIST(TIS$_0$) @ L=1
TIS_0_hist_mean [$/kWh]

FIG. 69

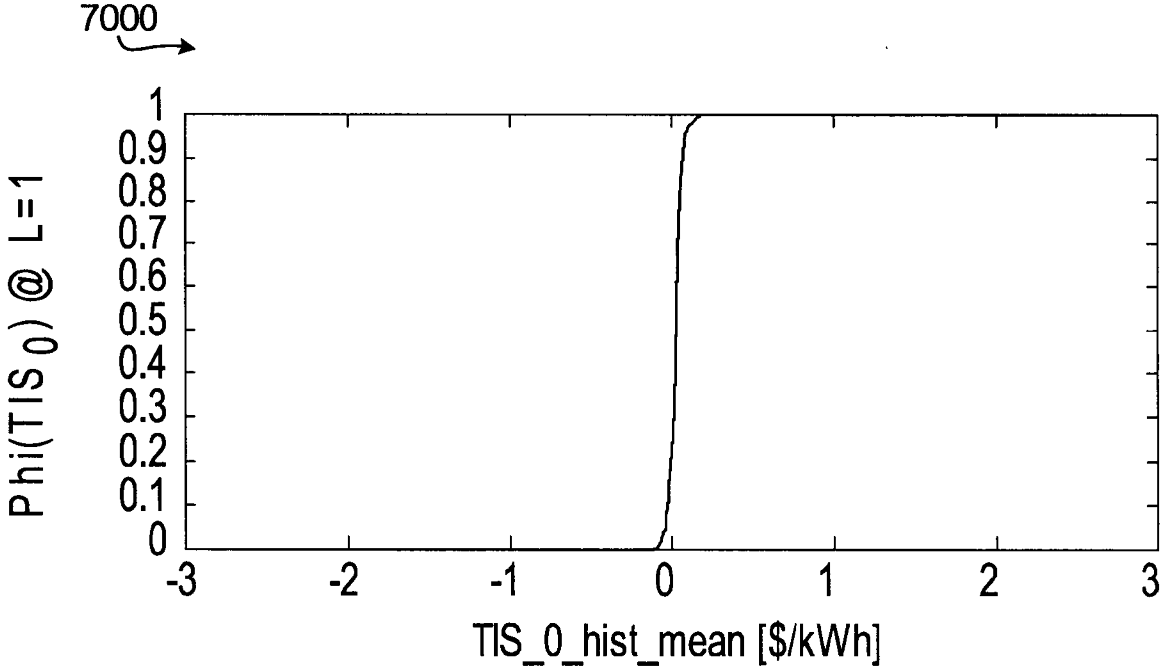

7000
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-3
-2
-1
0
1
2
3
Phi(TIS$_0$) @ L=1
TIS_0_hist_mean [$/kWh]

FIG. 70

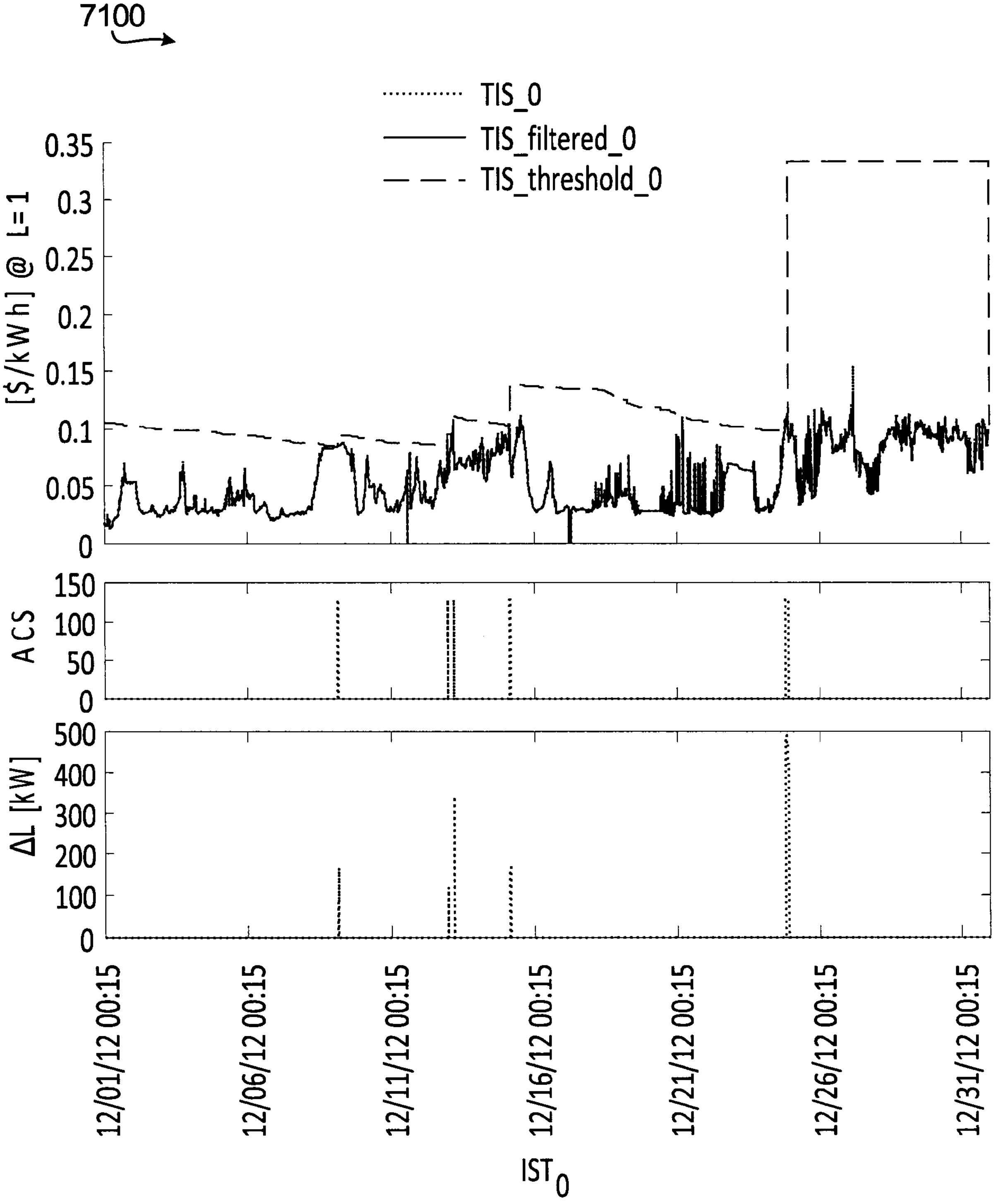

7100
TIS_0
TIS_filtered_0
TIS_threshold_0
[$/kWh] @ L=1
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
ACS
150
100
50
0
ΔL [kW]
500
400
300
200
100
0
12/01/12 00:15
12/06/12 00:15
12/11/12 00:15
12/16/12 00:15
12/21/12 00:15
12/26/12 00:15
12/31/12 00:15
IST0

FIG. 71

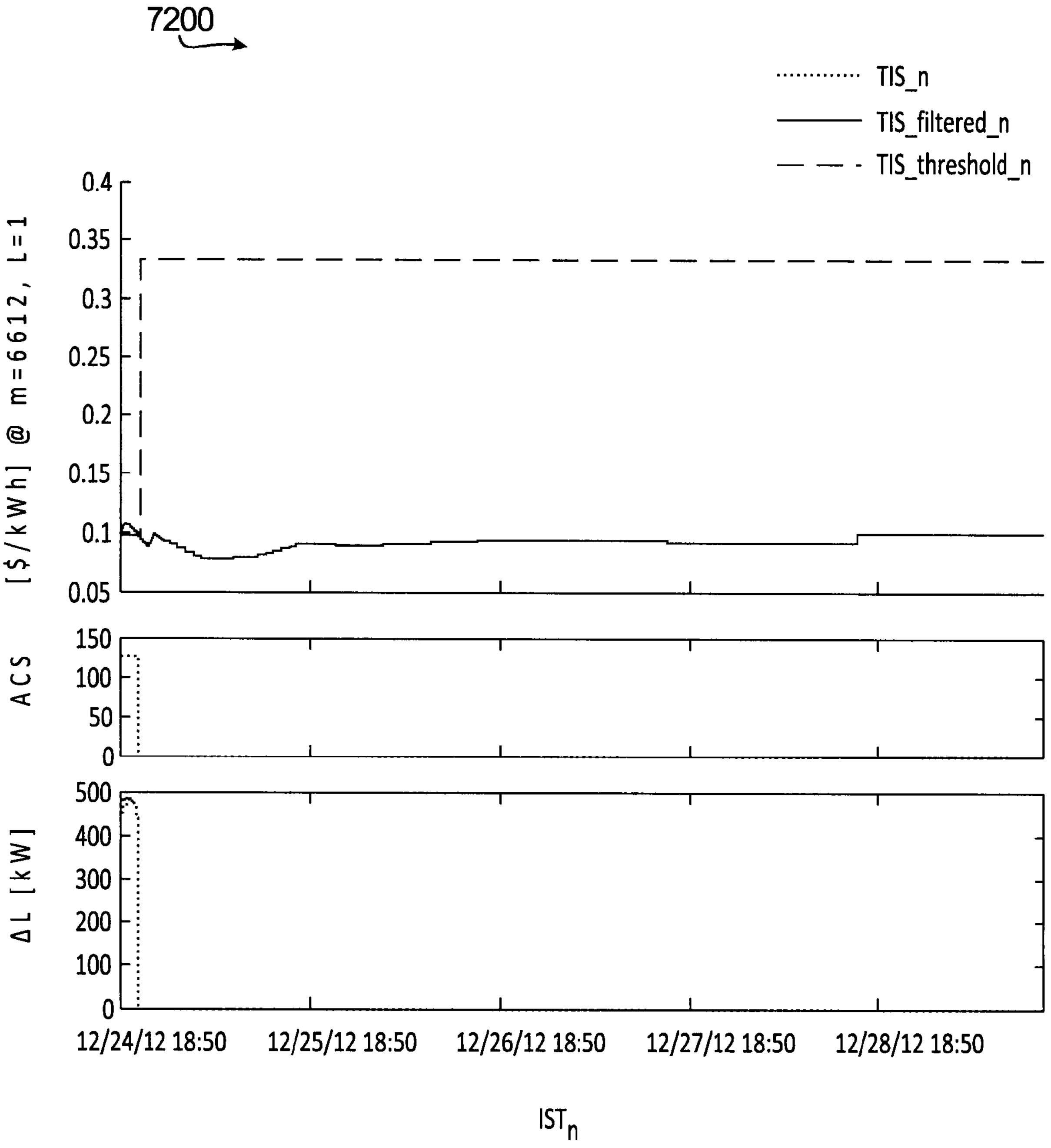

7200
TIS_n
TIS_filtered_n
TIS_threshold_n
[$/kWh] @ m=6612, L=1
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
ACS
150
100
50
0
ΔL [kW]
500
400
300
200
100
0
12/24/12 18:50
12/25/12 18:50
12/26/12 18:50
12/27/12 18:50
12/28/12 18:50
IST_n

FIG. 72

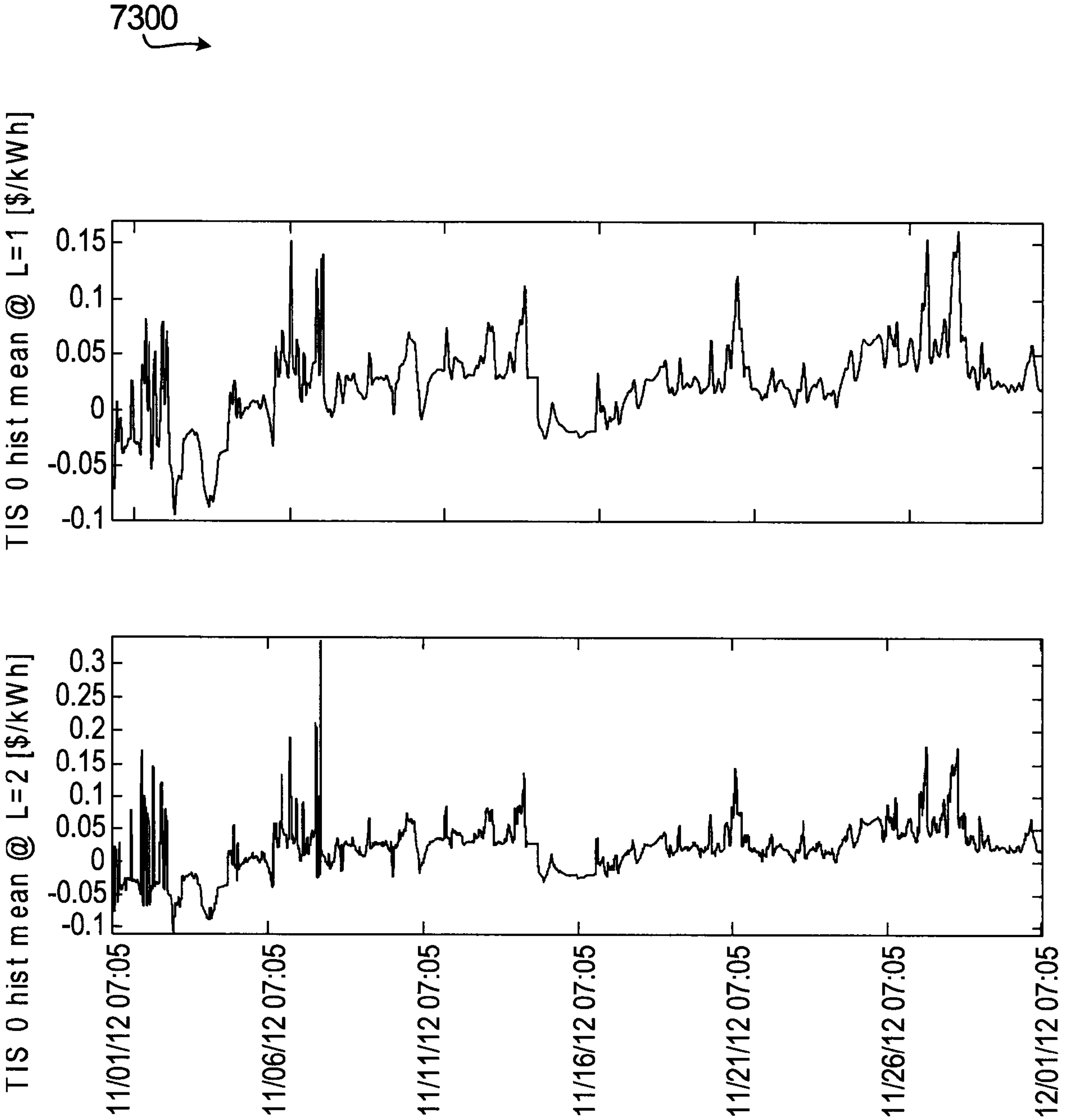

7300
TIS 0 hist mean @ L=1 [$/kWh]
0.15
0.1
0.05
0
-0.05
-0.1
TIS 0 hist mean @ L=2 [$/kWh]
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
11/01/12 07:05
11/06/12 07:05
11/11/12 07:05
11/16/12 07:05
11/21/12 07:05
11/26/12 07:05
12/01/12 07:05
UTC Time

FIG. 73

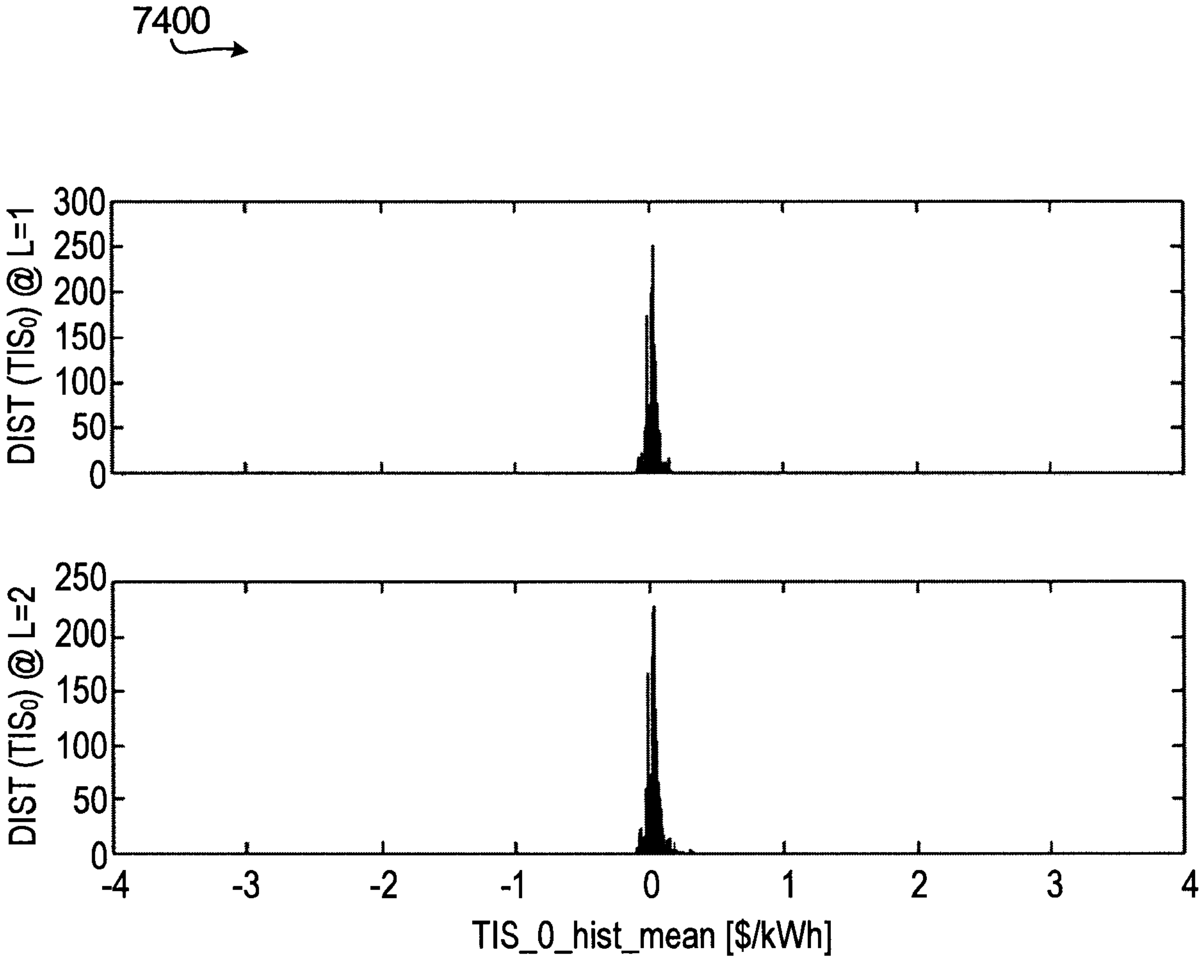

7400
DIST (TIS₀) @ L=1
300
250
200
150
100
50
0
DIST (TIS₀) @ L=2
250
200
150
100
50
0
-4
-3
-2
-1
0
1
2
3
4
TIS_0_hist_mean [$/kWh]

FIG. 74

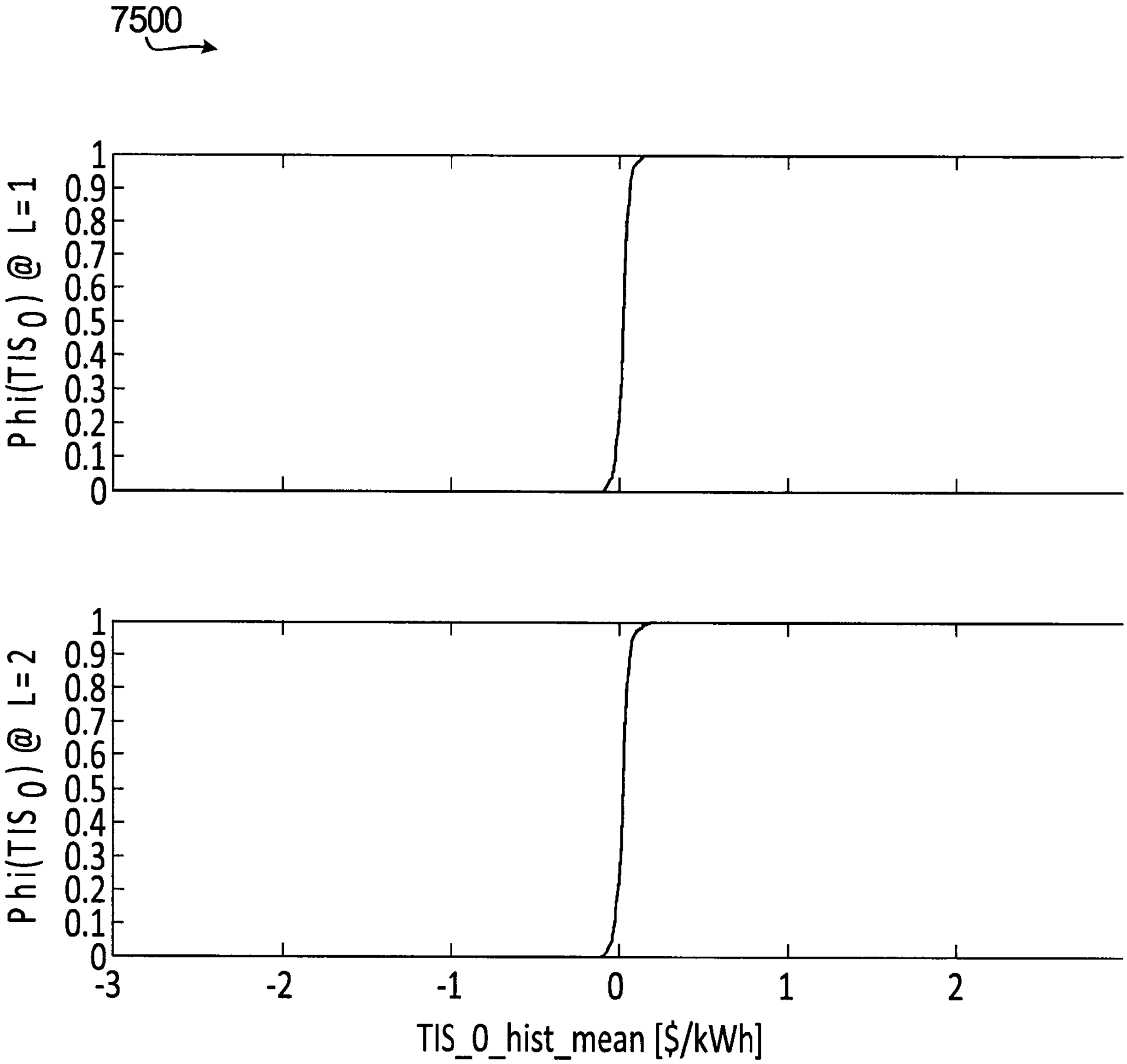
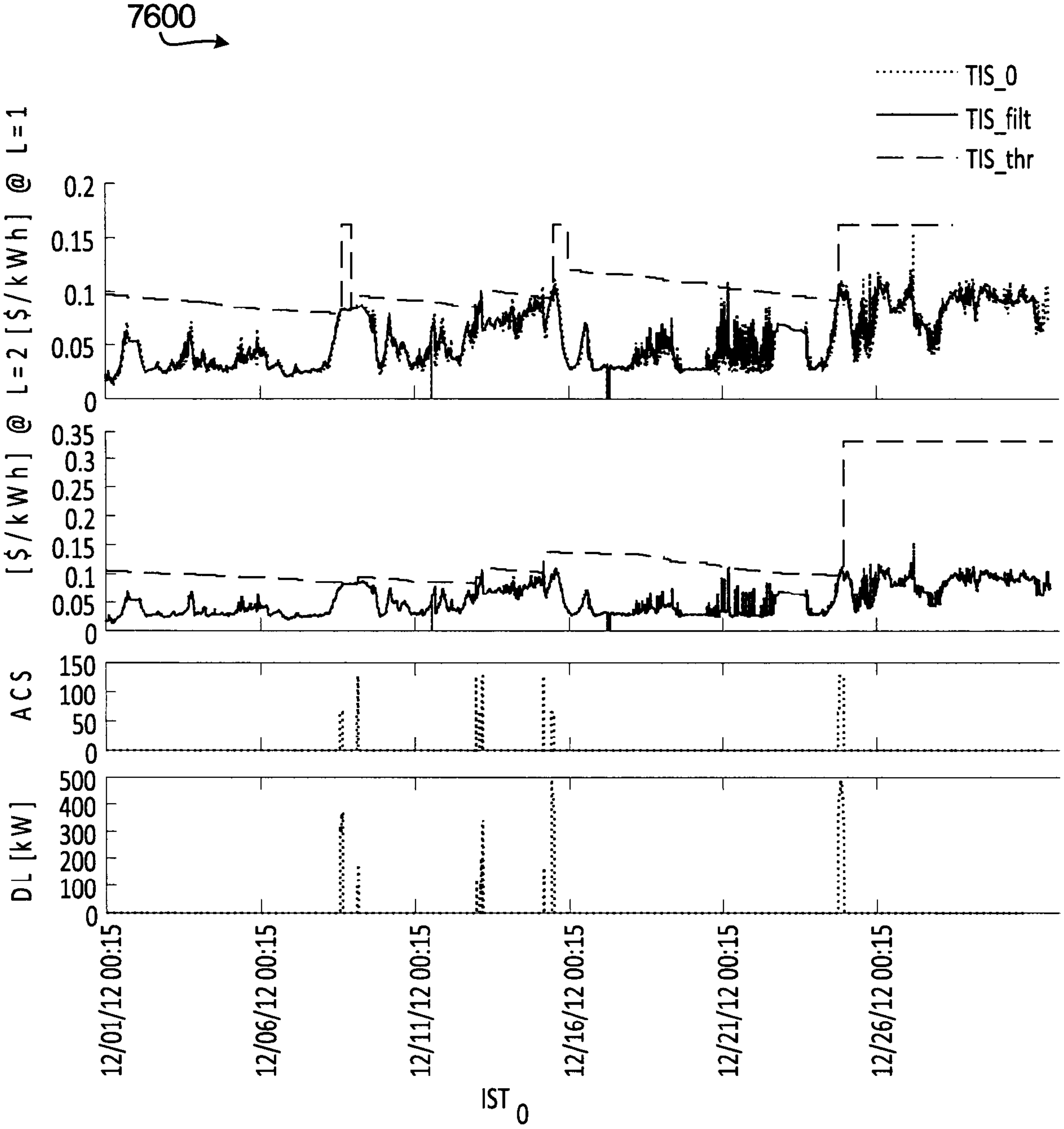

7500
Phi(TIS 0) @ L=1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Phi(TIS 0) @ L=2
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-3
-2
-1
0
1
2
TIS_0_hist_mean [$/kWh]

FIG. 75



7600
TIS_0
TIS_filt
TIS_thr
L=2 [$/kWh] @ L=1
[$/kWh] @
0.2
0.15
0.1
0.05
0
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
ACS
150
100
50
0
DL [kW]
500
400
300
200
100
0
12/01/12 00:15
12/06/12 00:15
12/11/12 00:15
12/16/12 00:15
12/21/12 00:15
12/26/12 00:15
IST_0

FIG. 76

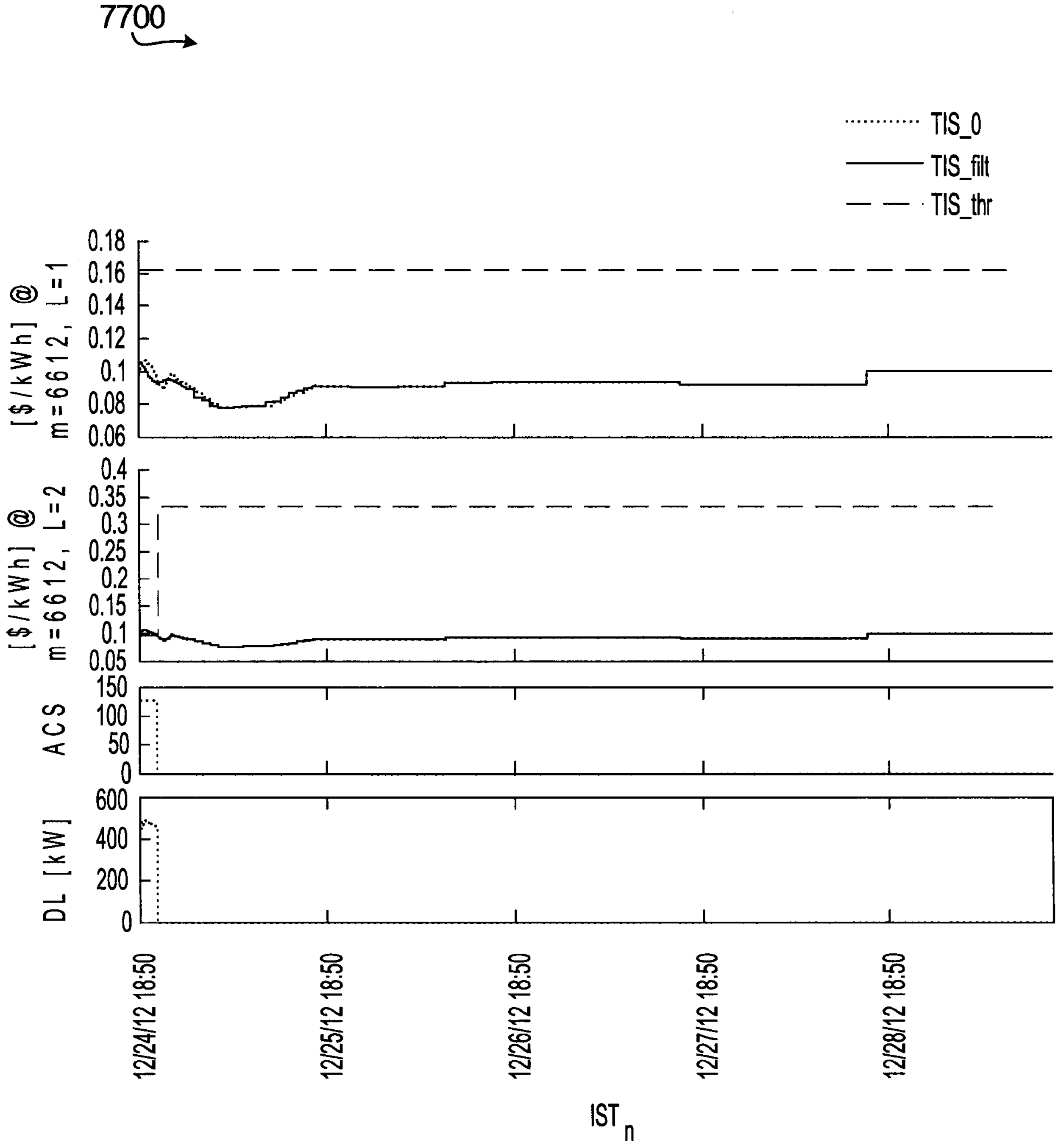

7700
TIS_0
TIS_filt
TIS_thr
[$/kWh] @ m=6612, L=1
0.18
0.16
0.14
0.12
0.1
0.08
0.06
[$/kWh] @ m=6612, L=2
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
ACS
150
100
50
0
DL [kW]
600
400
200
0
12/24/12 18:50
12/25/12 18:50
12/26/12 18:50
12/27/12 18:50
12/28/12 18:50
IST_n

FIG. 77

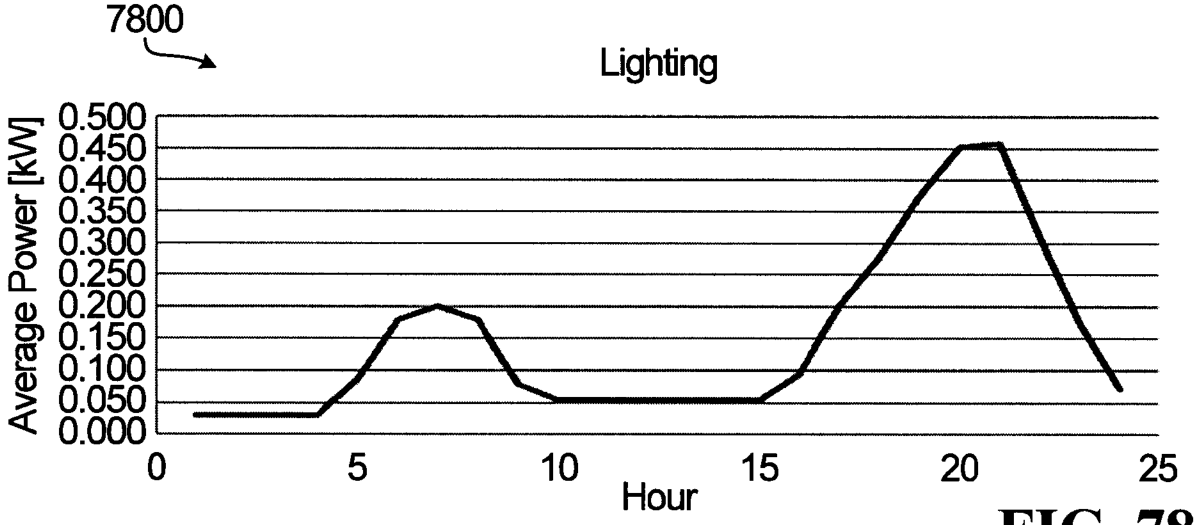


FIG. 78

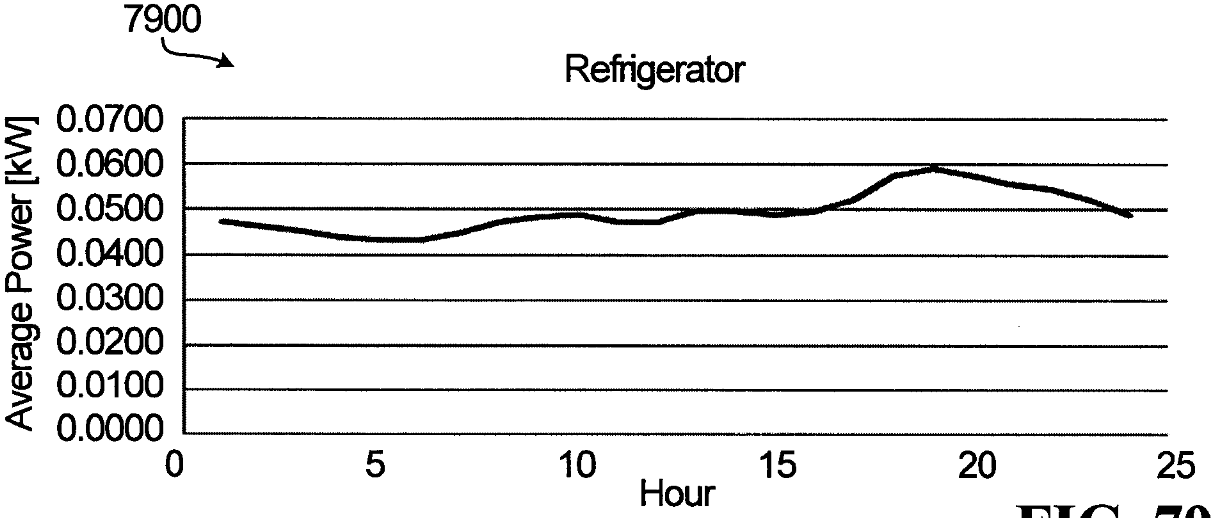


FIG. 79

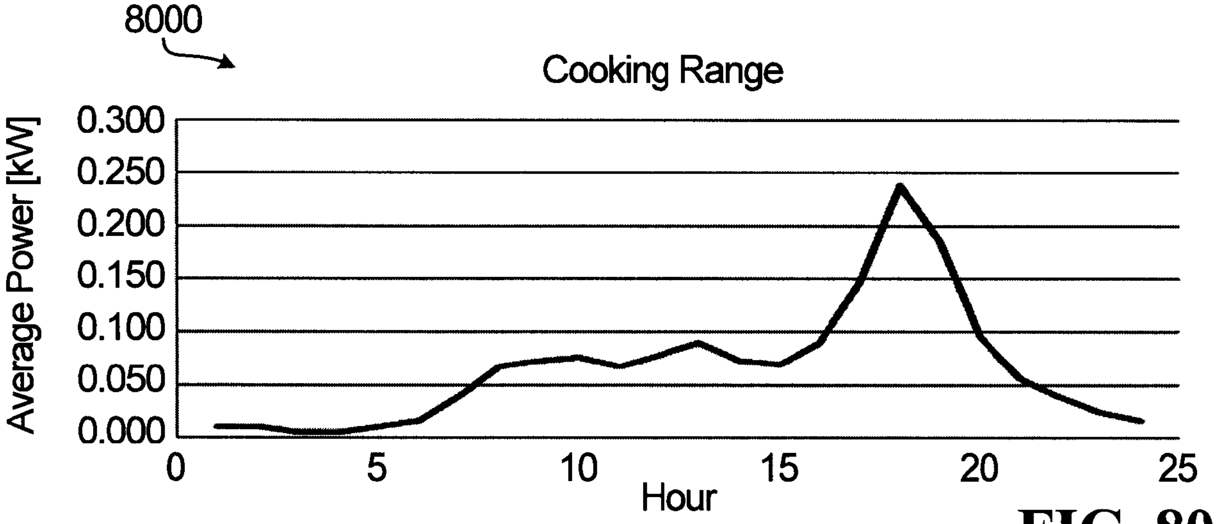


FIG. 80

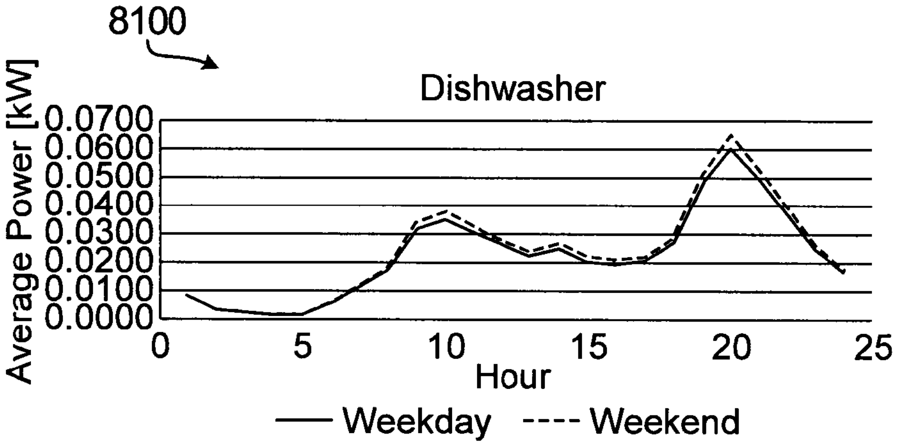


FIG. 81

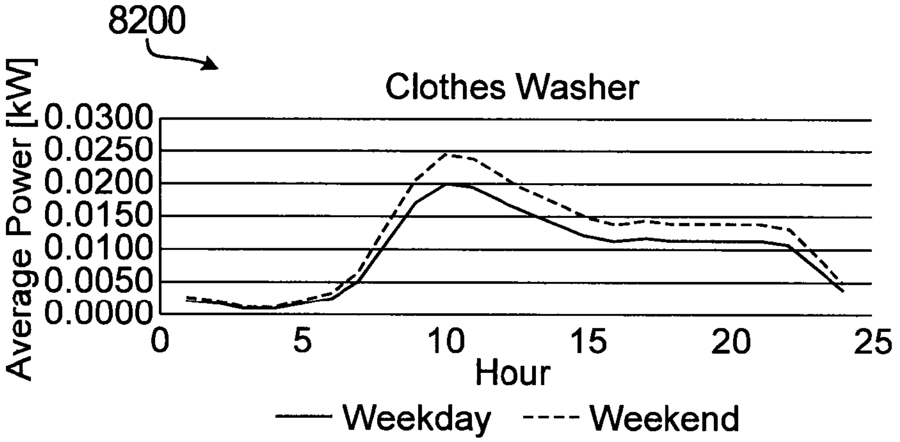


FIG. 82

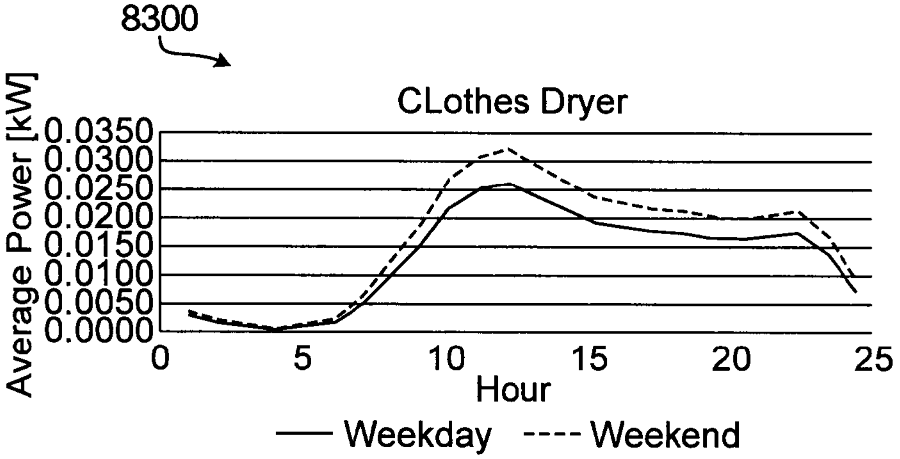


FIG. 83

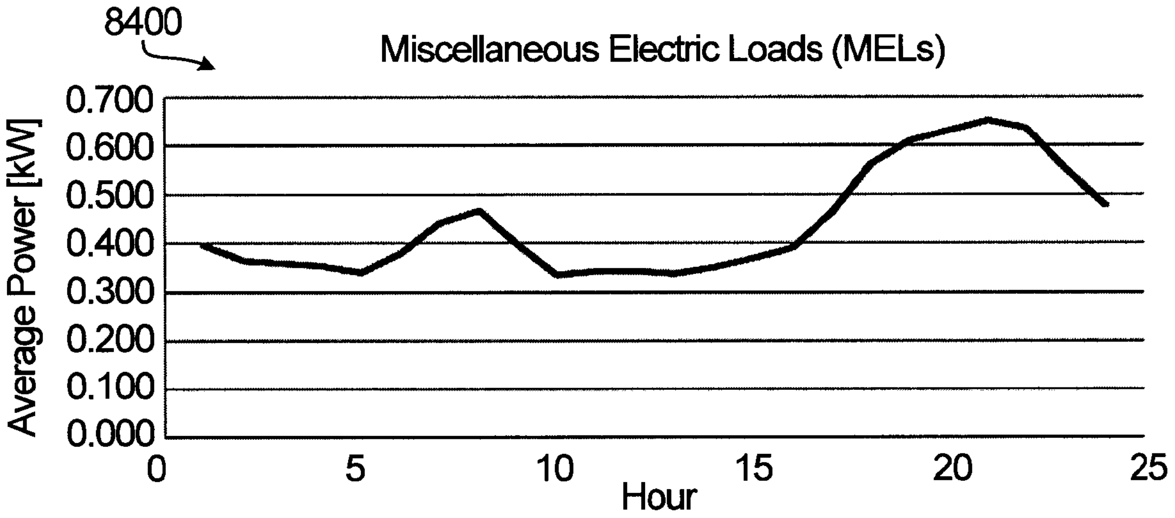

8400
Miscellaneous Electric Loads (MELs)
Average Power [kW]
0.700
0.600
0.500
0.400
0.300
0.200
0.100
0.000
0
5
10
15
20
25
Hour

FIG. 84

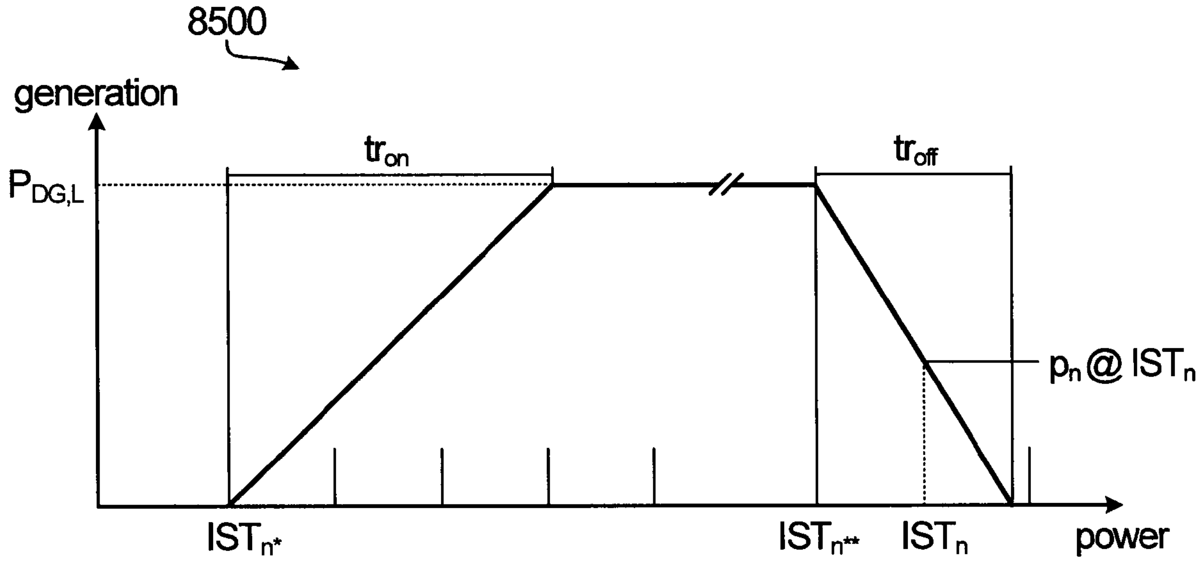

8500
generation
$P_{DG,L}$
tr_{on}
tr_{off}
p_n @ IST_n
IST_{n^*}
$IST_{n^{**}}$
IST_n
power

FIG. 85

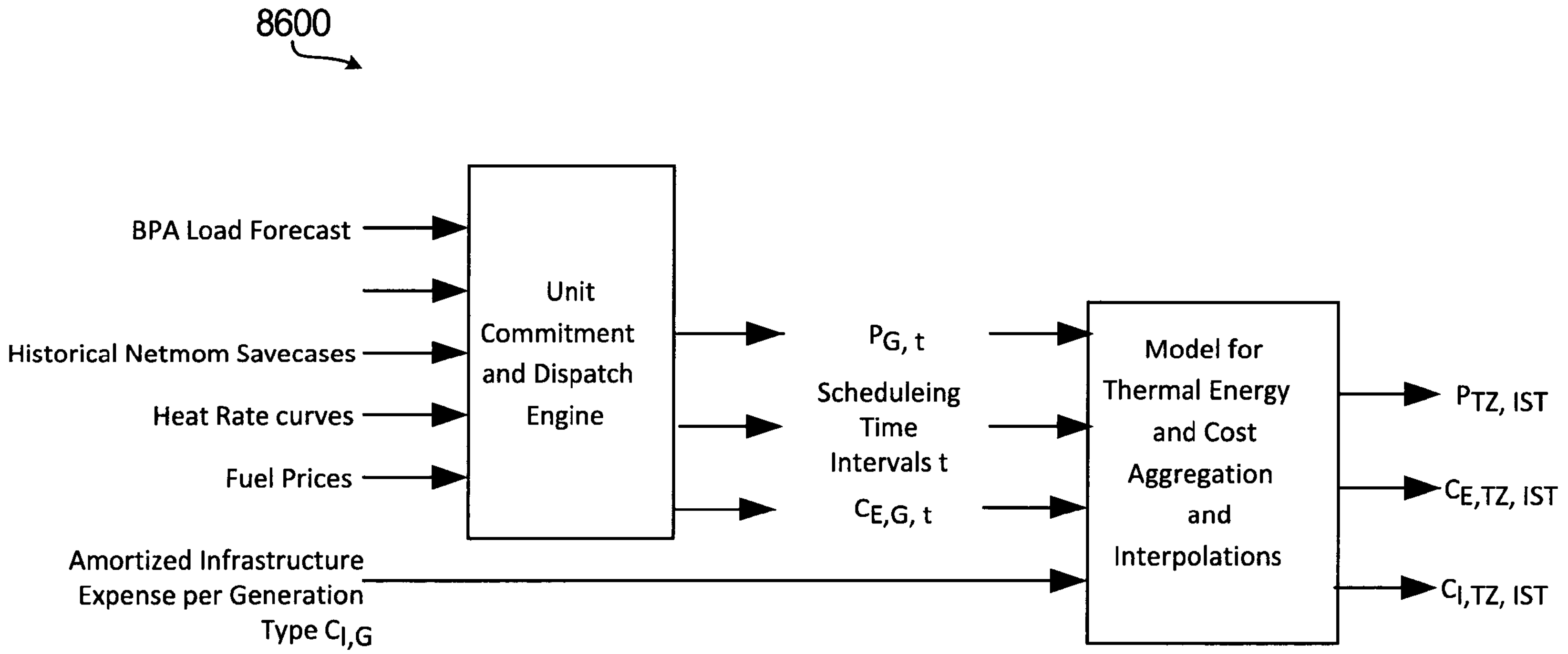

8600
BPA Load Forecast
Historical Netmom Savecases
Heat Rate curves
Fuel Prices
Amortized Infrastructure Expense per Generation Type $C_{I,G}$
Unit Commitment and Dispatch Engine
$P_{G,t}$
Scheduleing Time Intervals t
$C_{E,G,t}$
Model for Thermal Energy and Cost Aggregation and Interpolations
$P_{TZ,IST}$
$C_{E,TZ,IST}$
$C_{I,TZ,IST}$

FIG. 86

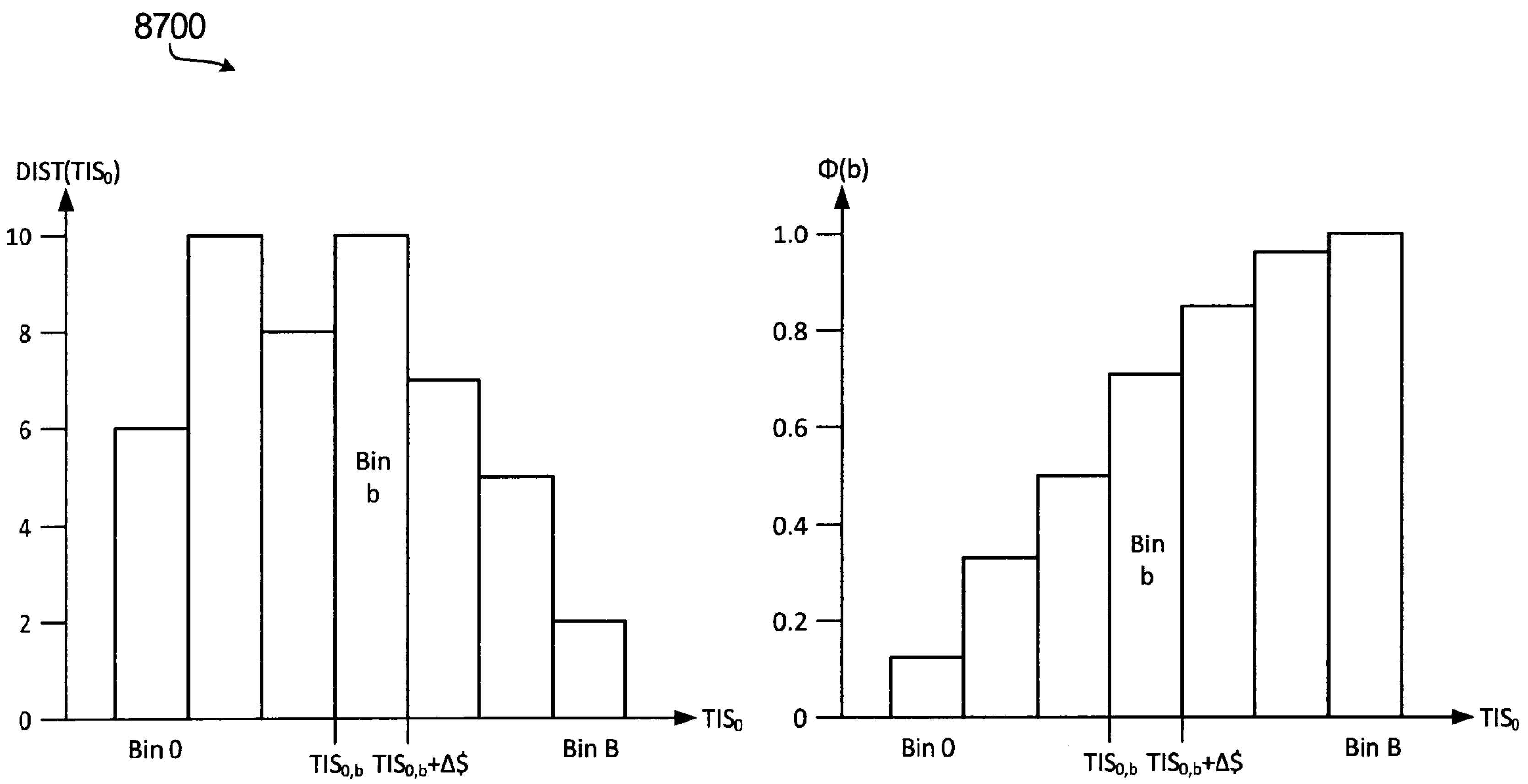

8700
DIST(TIS$_0$)
10
8
6
4
2
0
Bin 0
Bin b
TIS$_{0,b}$
TIS$_{0,b}$+Δ$
Bin B
TIS$_0$
Φ(b)
1.0
0.8
0.6
0.4
0.2
0
Bin 0
Bin b
TIS$_{0,b}$
TIS$_{0,b}$+Δ$
Bin B
TIS$_0$

FIG. 87

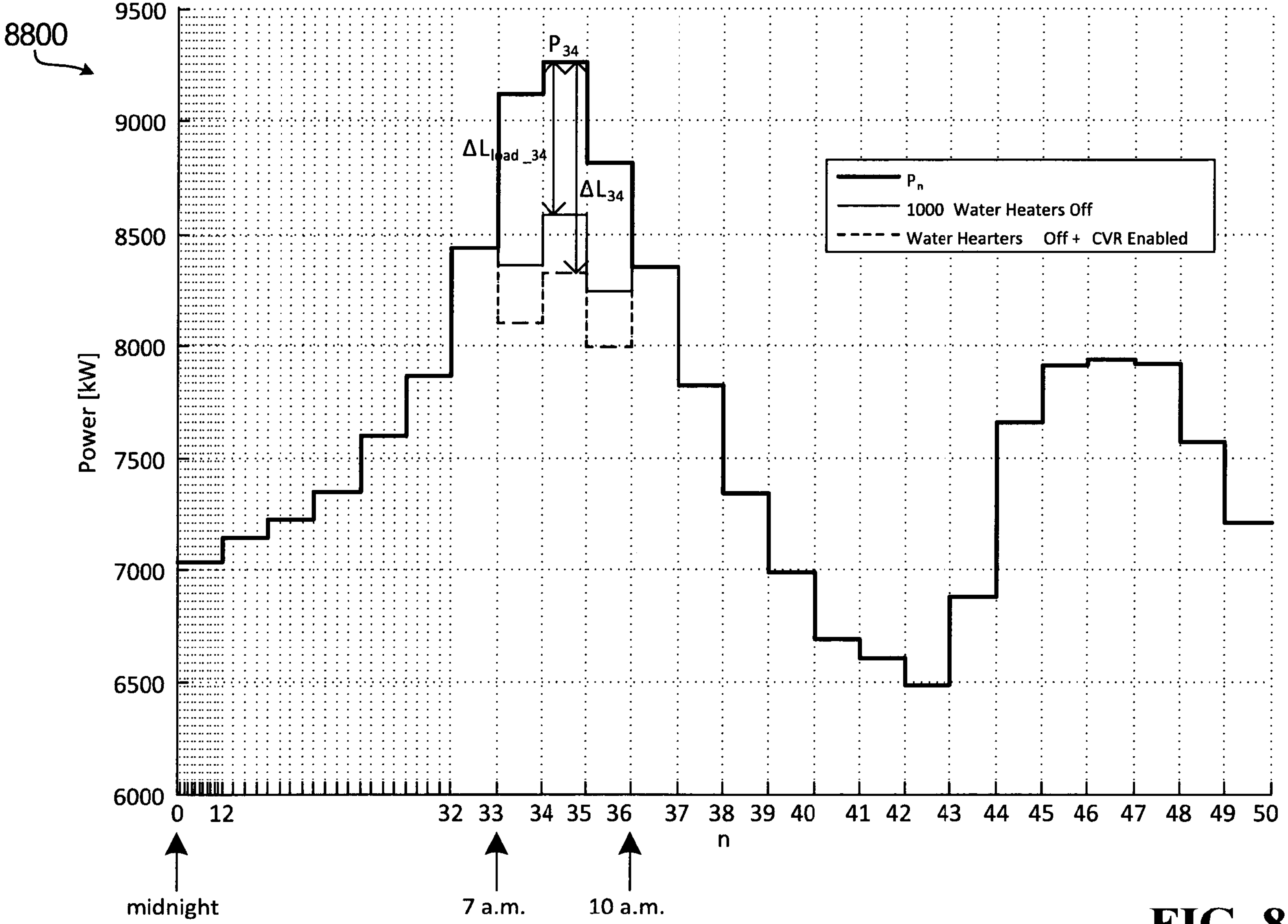

8800
9500
9000
8500
8000
7500
7000
6500
6000
Power [kW]
0
12
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
n
midnight
7 a.m.
10 a.m.
P_{34}
ΔL_{load_34}
ΔL_{34}
P_n
1000 Water Heaters Off
Water Hearters Off + CVR Enabled

FIG. 88

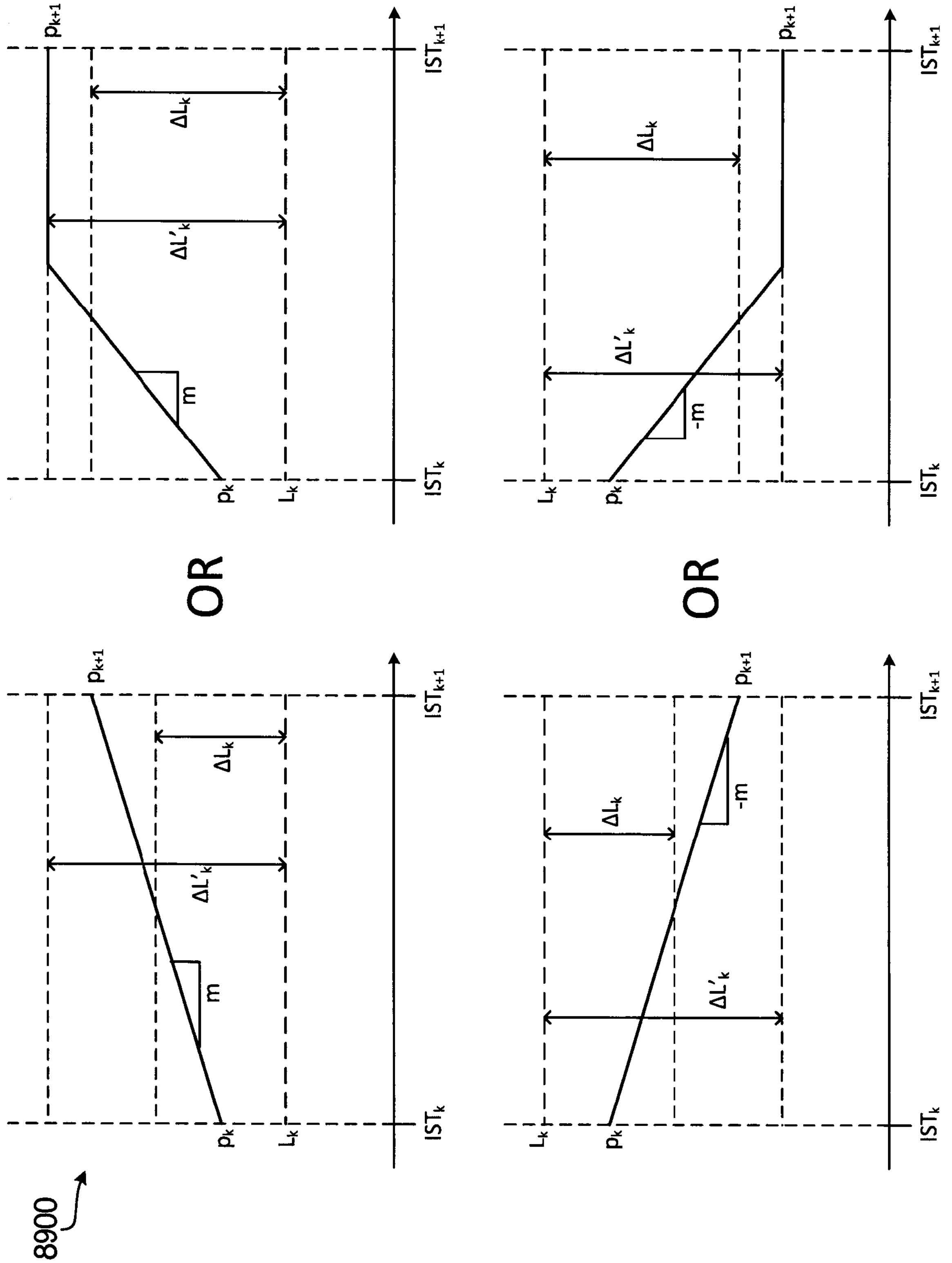


FIG. 89

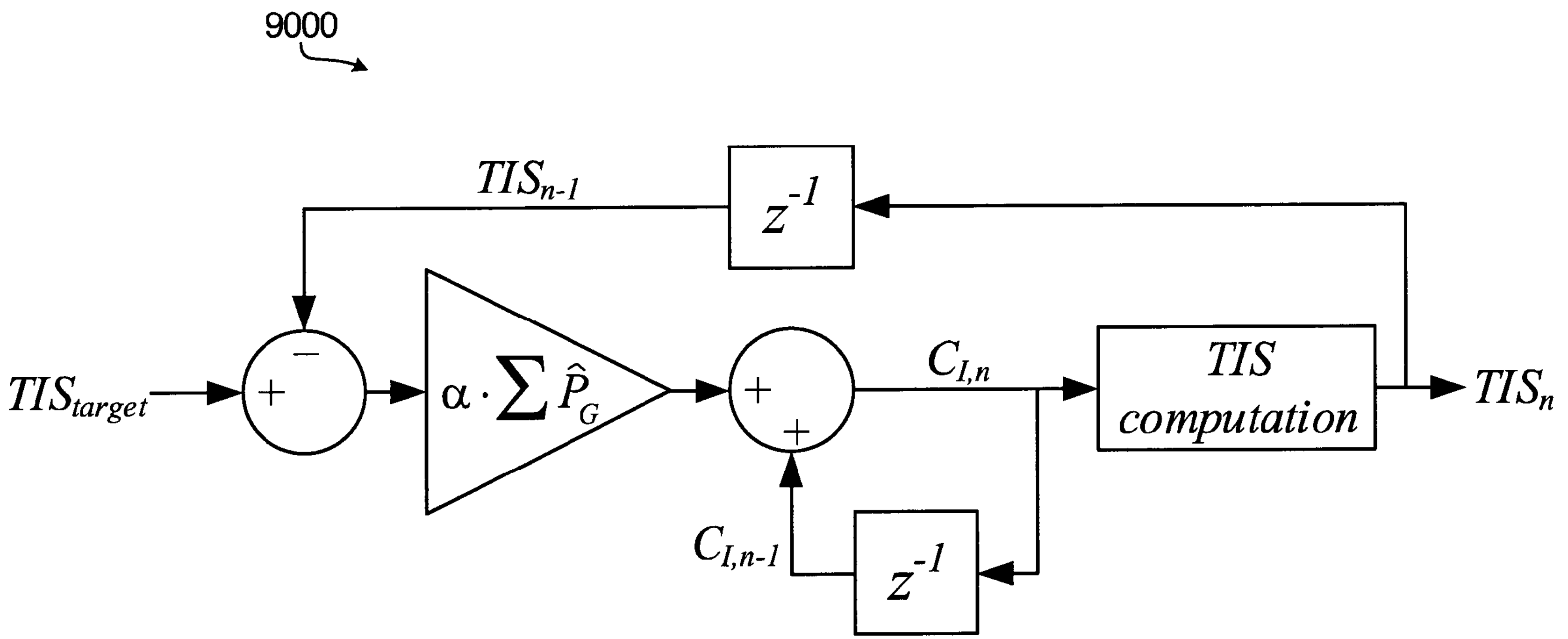

9000
TIS_{target}
+
−
TIS_{n-1}
z^{-1}
$\alpha \cdot \sum \hat{P}_G$
+
+
$C_{I,n}$
$C_{I,n-1}$
z^{-1}
TIS computation
TIS_n

FIG. 90

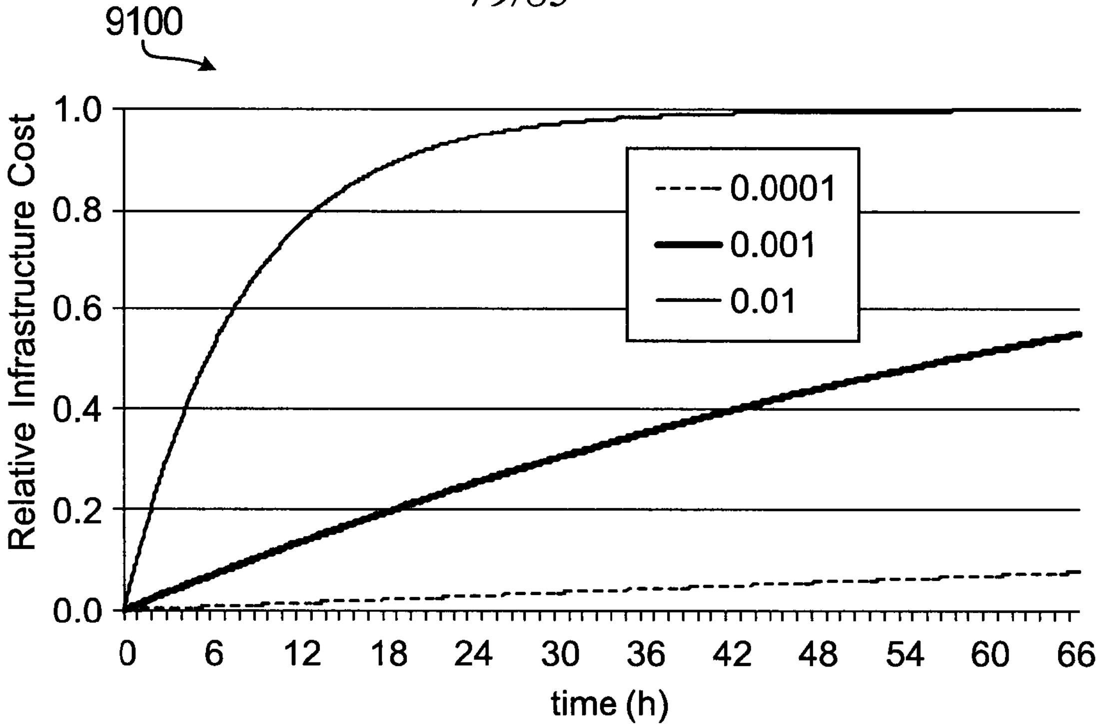

9100
Relative Infrastructure Cost
1.0
0.8
0.6
0.4
0.2
0.0
0
6
12
18
24
30
36
42
48
54
60
66
time (h)
0.0001
0.001
0.01

FIG. 91

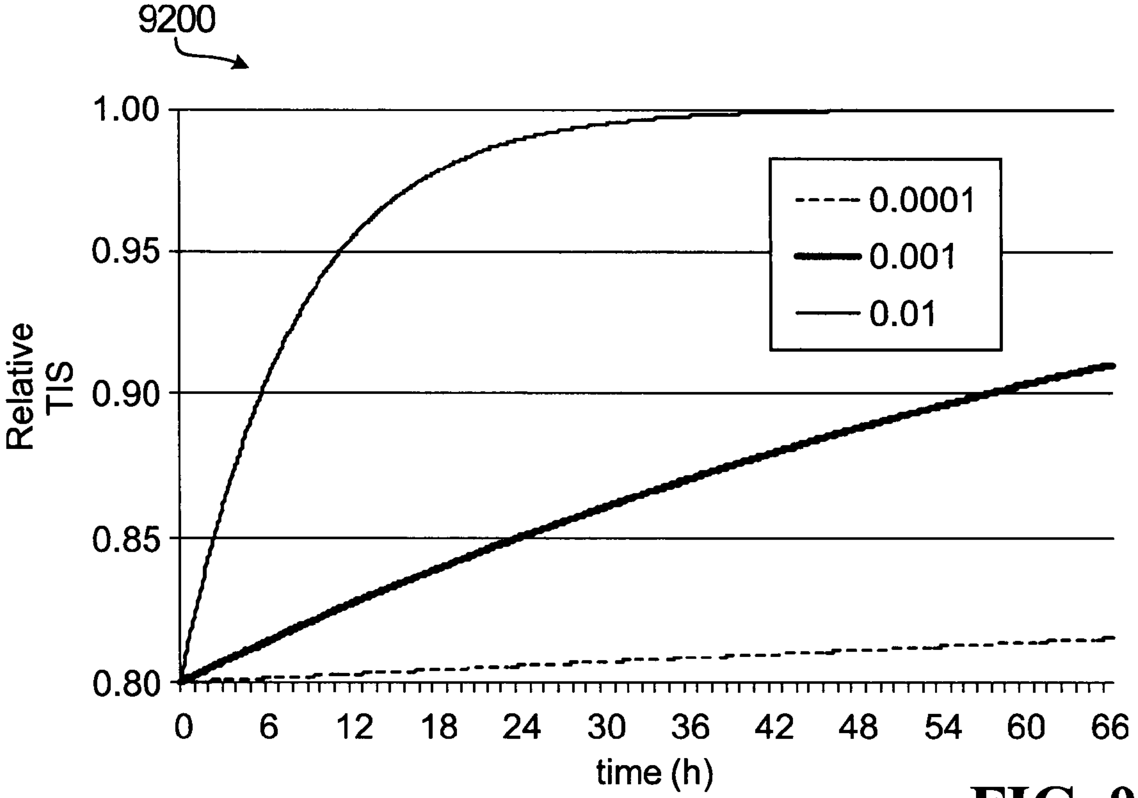

9200
Relative TIS
1.00
0.95
0.90
0.85
0.80
0
6
12
18
24
30
36
42
48
54
60
66
time (h)
0.0001
0.001
0.01

FIG. 92

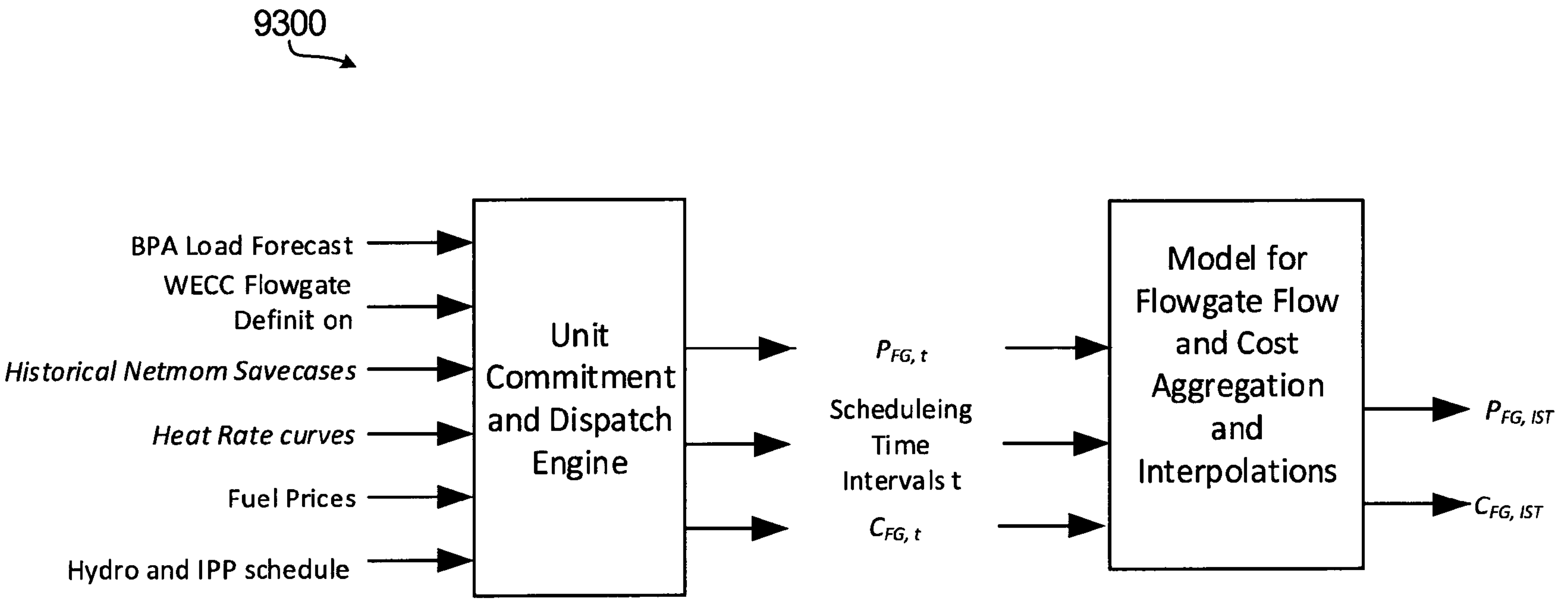

9300
BPA Load Forecast
WECC Flowgate Definit on
Historical Netmom Savecases
Heat Rate curves
Fuel Prices
Hydro and IPP schedule
Unit Commitment and Dispatch Engine
$P_{FG,t}$
Scheduleing Time Intervals t
$C_{FG,t}$
Model for Flowgate Flow and Cost Aggregation and Interpolations
$P_{FG,IST}$
$C_{FG,IST}$

FIG. 93

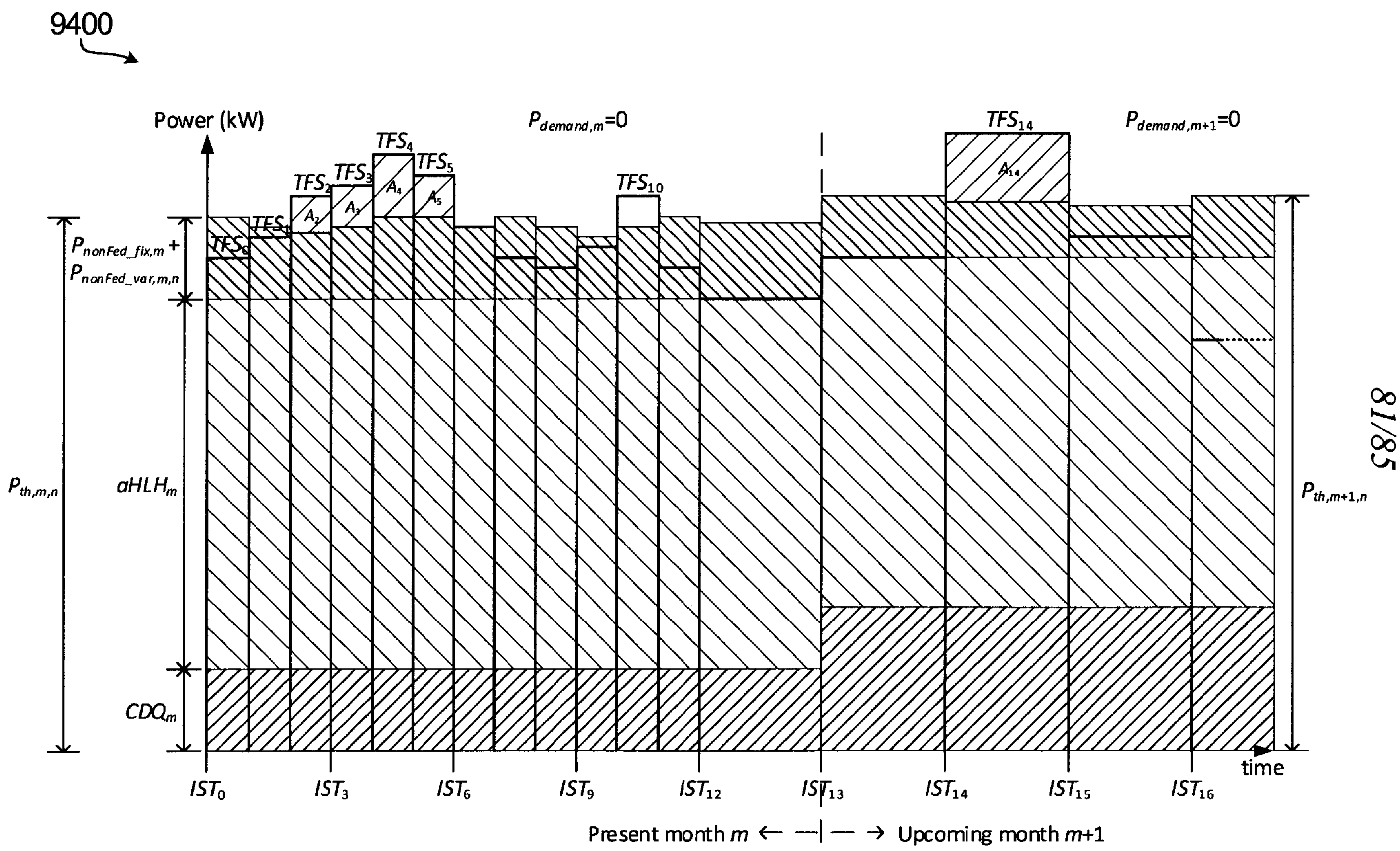

9400
Power (kW)
$P_{demand,m}$=0
$P_{demand,m+1}$=0
TFS_0
TFS_1
TFS_2
TFS_3
TFS_4
TFS_5
TFS_{10}
TFS_{14}
A_2
A_3
A_4
A_5
A_{14}
$P_{nonFed\_fix,m}$ + $P_{nonFed\_var,m,n}$
$aHLH_m$
CDQ_m
$P_{th,m,n}$
$P_{th,m+1,n}$
time
IST_0
IST_3
IST_6
IST_9
IST_{12}
IST_{13}
IST_{14}
IST_{15}
IST_{16}
Present month m ← —|— → Upcoming month m+1

FIG. 94

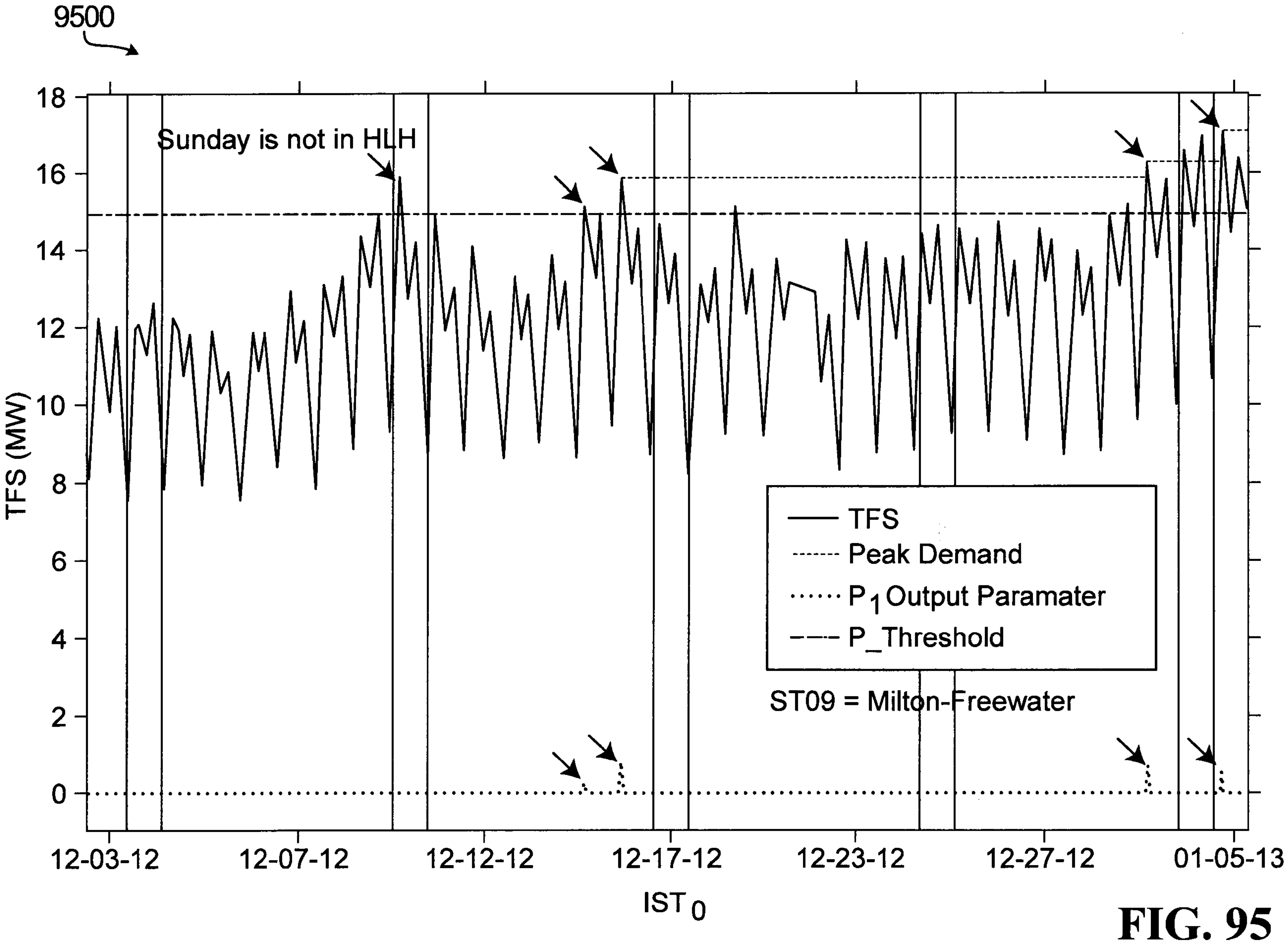

9500
TFS (MW)
18
16
14
12
10
8
6
4
2
0
Sunday is not in HLH
TFS
Peak Demand
P_1 Output Paramater
P_Threshold
ST09 = Milton-Freewater
12-03-12
12-07-12
12-12-12
12-17-12
12-23-12
12-27-12
01-05-13
IST_0

FIG. 95

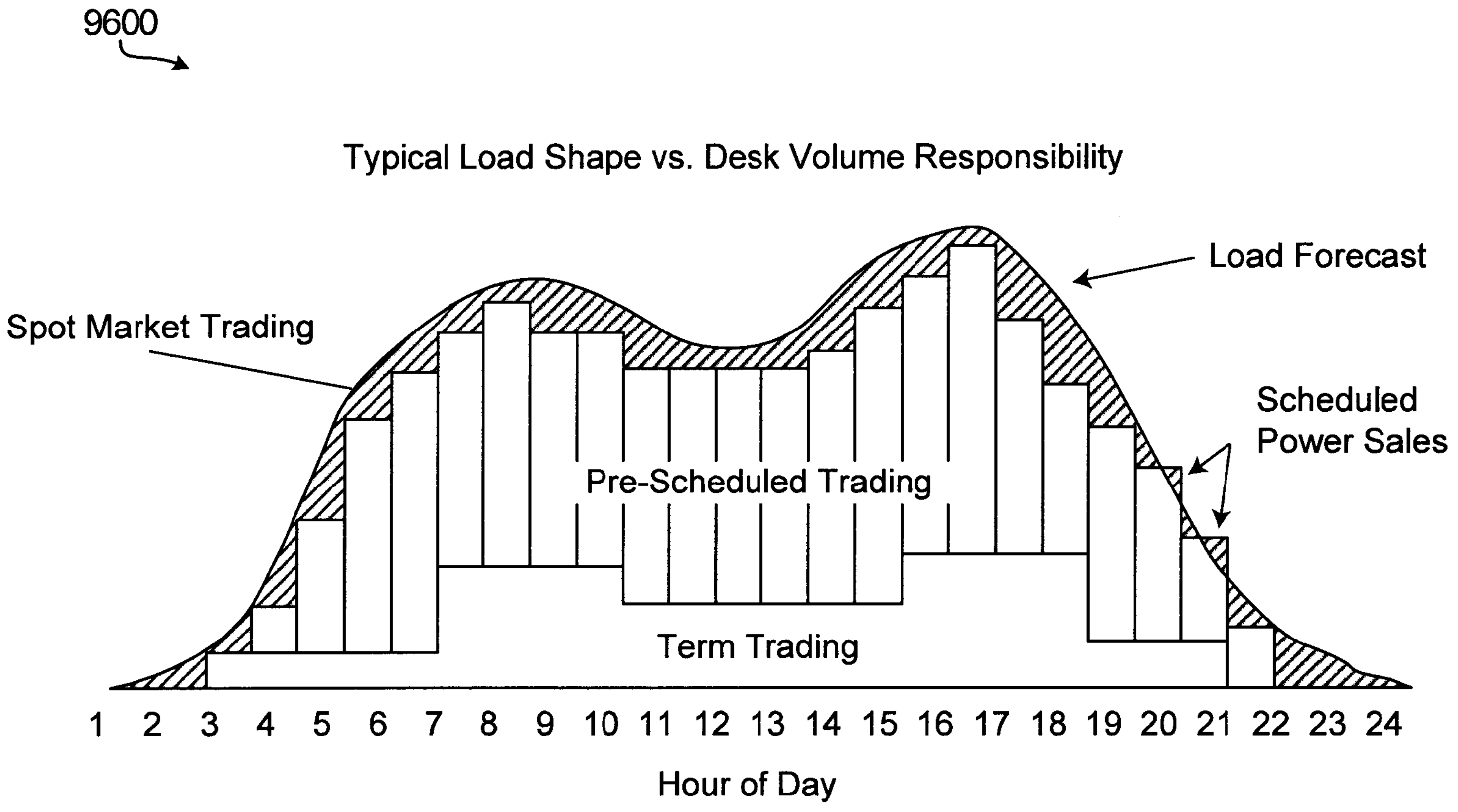

9600
Typical Load Shape vs. Desk Volume Responsibility
Spot Market Trading
Load Forecast
Scheduled
Power Sales
Pre-Scheduled Trading
Term Trading
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Hour of Day

FIG. 96

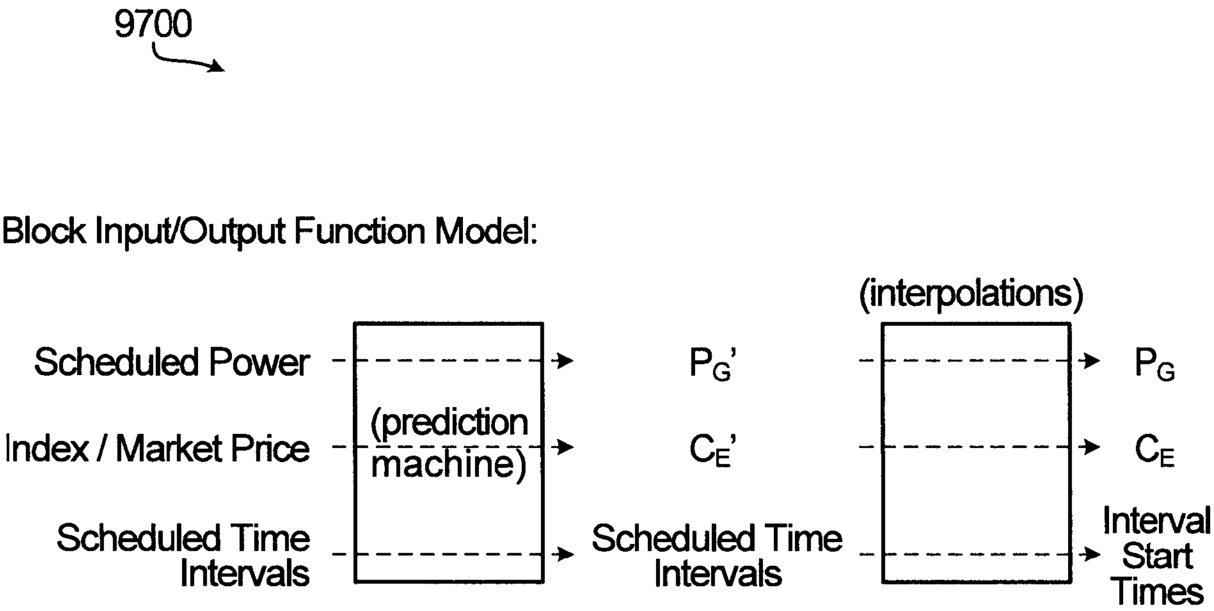

9700
Block Input/Output Function Model:
Scheduled Power
Index / Market Price
Scheduled Time Intervals
(prediction machine)
P_G'
C_E'
Scheduled Time Intervals
(interpolations)
P_G
C_E
Interval Start Times

FIG. 97

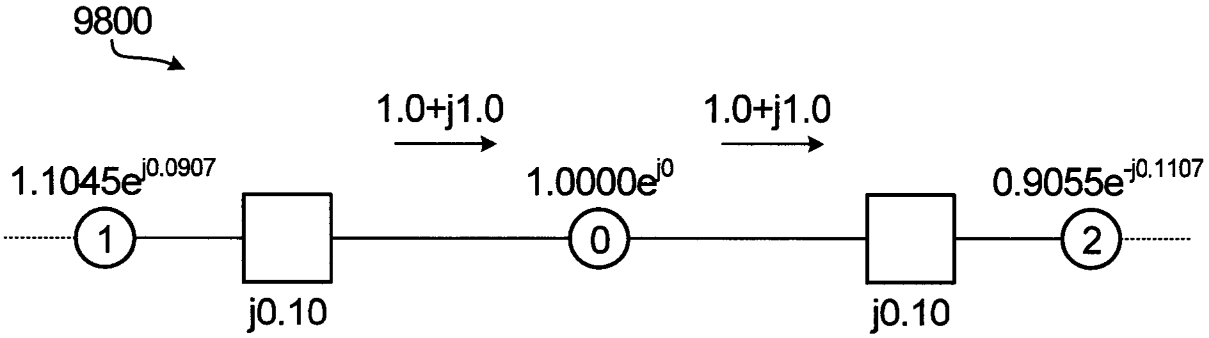


FIG. 98

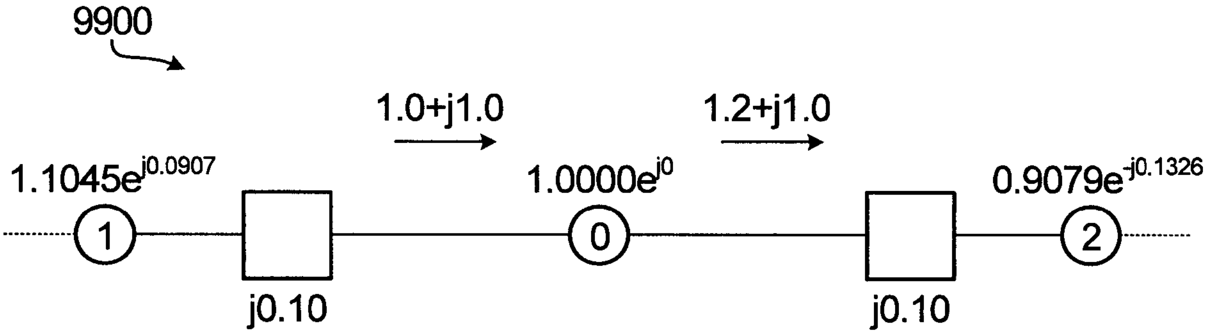


FIG. 99

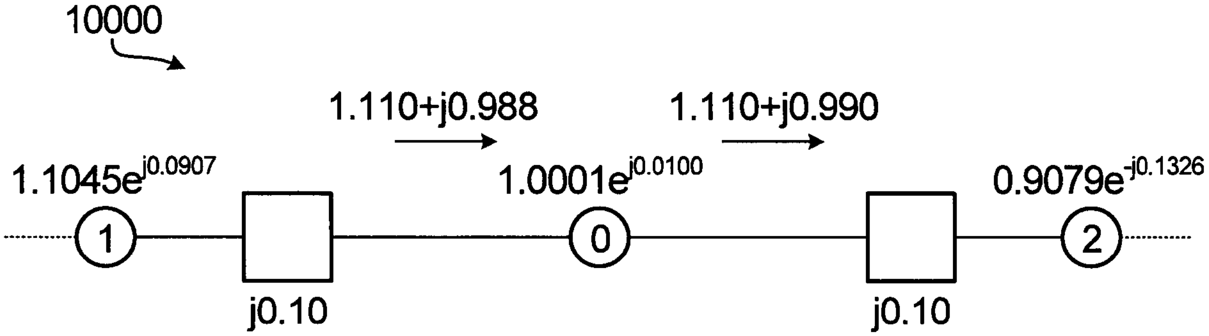
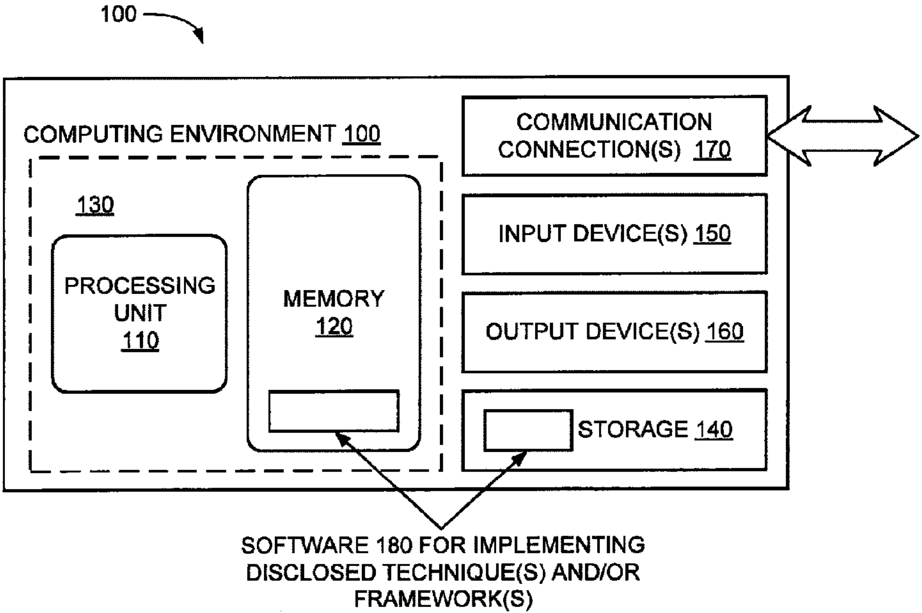


FIG. 100


100
COMPUTING ENVIRONMENT 100
130
PROCESSING UNIT 110
MEMORY 120
COMMUNICATION CONNECTION(S) 170
INPUT DEVICE(S) 150
OUTPUT DEVICE(S) 160
STORAGE 140
SOFTWARE 180 FOR IMPLEMENTING DISCLOSED TECHNIQUE(S) AND/OR FRAMEWORK(S)